2-19-2013

Imaging methods for understanding and improving visual training in the geosciences

Brandon May

Imaging Methods for Understanding and Improving Visual Training in the Geosciences

by

Brandon B. May

B.A. Mathematics & Physics
Skidmore College, 2008

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science

Chester F. Carlson Center for Imaging Science
College of Science
Rochester Institute of Technology
Rochester, New York

February 19, 2013

Brandon B. May Date

Coordinator, M.S. Degree Program Date

Chester F. Carlson Center For Imaging Science
College of Science
Rochester Institute Of Technology
Rochester, New York

Certificate of Approval

M.S. Degree Thesis

The M.S. degree thesis of Brandon B. May
has been examined and approved by the
thesis committee as satisfactory for the
thesis required for the M.S. degree in Imaging Science

Dr. Nathan D. Cahill, Thesis Advisor — Date

Dr. Jeff B. Pelz, Committee Member — Date

Dr. Harvey E. Rhody, Committee Member — Date

I dedicate this thesis to M. P., M. M., & S. M.

Acknowledgements

This research project would not have been possible without the support of many people. I'd like to thank my advisor Nate Cahill and the rest of my thesis committee, Jeff Pelz and Harvey Rhody, for their support and helpful suggestions in our weekly meetings as I discussed my progress and developed my algorithms. Thanks also to Nate for providing valuable feedback on the initial drafts of my thesis.

Thanks to Mitch Rosen for helping me to get started and involved with the GeoVis research project, and thanks for the efforts of Mitch, Jeff, John Tarduno, and Robbie Jacobs who first got this project going.

For their great help while out imaging in the field I'd like to thank Dave Kelbe and Tommy Keane. Each of them proved invaluable in assisting me to capture many panoramas and keeping me entertained during the long drives through California. I'd also like to thank Karen Evans, Piyush Agarwal, and Jason Babcock for their work capturing the eye-tracking data during the field trips and for providing me with good company throughout our camping adventures. Karen and Piyush also spent many hours with me in the CSI getting the projector system working well, for which I am very appreciative.

My ground truth analysis wouldn't have been possible without the help of Tommy and a lot of manual clicking labor done by Sarah Piper, so thanks very much for that!

Finally, I'd like to thank my family for being supportive of my studies at RIT and always encouraging me to do my best.

RIT, February 2013

Brandon B. May

Abstract

Experience in the field is a critical educational component of every student studying geology. However, it is typically difficult to ensure that every student gets the necessary experience because of monetary and scheduling limitations. Thus, we proposed to create a virtual field trip based off of an existing 10-day field trip to California taken as part of an undergraduate geology course at the University of Rochester. To assess the effectiveness of this approach, we also proposed to analyze the learning and observation processes of both students and experts during the real and virtual field trips.

At sites intended for inclusion in the virtual field trip, we captured gigapixel resolution panoramas by taking hundreds of images using custom built robotic imaging systems. We gathered data to analyze the learning process by fitting each geology student and expert with a portable eye-tracking system that records a video of their eye movements and a video of the scene they are observing. An important component of analyzing the eye-tracking data requires mapping the gaze of each observer into a common reference frame. We have made progress towards developing a software tool that helps automate this procedure by using image feature tracking and registration methods to map the scene video frames from each eye-tracker onto a reference panorama for each site.

For the purpose of creating a virtual field trip, we have a large scale semi-immersive display system that consists of four tiled projectors, which have been colorimetrically and photometrically calibrated, and a curved widescreen display surface. We use this system to present the previously captured panoramas, which simulates the experience of visiting the sites in person. In terms of broader geology education and outreach, we have created an interactive website that uses Google Earth as the interface for visually exploring the panoramas captured for each site.

Contents

List of Figures

Chapter 1

Introduction

Experience in the field is a critical educational component of every student studying geology. However, it is typically difficult to ensure that every student gets the necessary experience because of monetary and scheduling limitations. If anything, the field trip will be restricted to a single-day with a small group of people. Perhaps in these cases it would be effective to bring the field experience to the students instead of sending the students out into the field. Thus, we proposed to create a virtual field trip based off of an existing 10-day field trip to California taken as part of an undergraduate geology course at the University of Rochester [1]. During the ten-day field trip both geology students and expert geologists visit a number of geologic sites.

In order to create a virtual field trip that realistically simulates field experience, the original sites must be carefully captured in such a way that later allows for a realistic simulation. Therefore, at sites intended for inclusion in the virtual field trip, we captured gigapixel resolution panoramas by taking hundreds of images using custom built robotic imaging systems. The individual images were processed and stitched together at a later time to form high-resolution panoramic images. To create the virtual field trip the panoramas are then projected for observation in a semi-immersive display system.

An important part of this project is the ability to analyze the learning process during the real experience as compared to the virtual one as well as the educational effectiveness of each. Towards this goal, an active vision approach has been adopted as a means of analyzing student

learning based on where they look. When both in the field and using the semi-immersive display, each geology student is fit with a portable eye-tracking system that records a video of their eye movements and a video of the scene they are observing. Afterwards, these data can be processed to determine where the person was looking (their gaze) and for how long. To provide a baseline for comparison, several expert geologists were present during the real and virtual field trips to participate in the same experience as the students.

It is possible to compare the eye-tracking information from one observer to the next and from the same observer to themselves over a period of time to evaluate information such as novice-expert differences, the learning process through time, and differences in the learning experience between the real and virtual field trips. The ability to make these comparisons relies on having a common reference frame on which to compare the gaze data. Typically fixations are plotted manually from each observers scene video onto a panorama of the scene, however this is a time consuming process. We have made progress towards developing a software tool that helps automate this procedure by using video feature tracking and image registration methods to map gaze information from each video frame onto a reference panorama.

In terms of geology education and outreach, the panoramic images collected for the virtual field trip could be incorporated into a range of geoscience courses at every educational level. The database of images could even prove useful to researchers in the field of computer vision for testing feature detection, image stitching, structure from motion, and other techniques. Cognitive science educators and researchers could also find value in the large database of video eye-tracking data of novices and experts viewing geological landscapes. Towards these goals, we have created an interactive website that uses Google Earth as the interface for visually exploring the panoramas captured for each site [2]. It has the potential to incorporate additional data such as audio recordings, eye-tracking data, and other information relevant for use as an educational tool.

The rest of this thesis is focused on presenting the details of our work. In chapter 2 we present an overview of the prior work related to this research. We describe the tools and techniques for capturing gigapixel resolution panoramas out in the field in chapter 3. In chapter 4 we describe the tools and algorithms necessary to generate the panoramas and perform video frame to panorama registration. In chapter 5 we explain the setup and calibration of the semi-immersive virtual

environment and describe the online virtual field trip interface. We conclude and propose possible directions for future work in chapter 6.

Novel contributions are presented in each of the chapters three through five. The following summarizes these contributions according to each chapter:

Ch. 3: Imaging in the Field

- Built a portable gigapixel resolution robotic imaging system using a telescope mount, DSLR camera, battery pack, and wireless bluetooth control via tablet computer.
- Designed an efficient shooting pattern on the sphere to minimize time of capture.

Ch. 4: Image Processing

- Developed a two-pass SIFT feature tracking algorithm for videos.
- Developed a feature matching algorithm that used a multiple level-per-octave SIFT approach to be robust to very large differences in resolution and scale.
- Developed a video keyframe selection algorithm based on initial video frame to panorama homography computations.
- Developed a method to match video frames to a reference panorama by calculating a homography by combining information from SIFT video tracking, multi-level feature matching, and keyframe selection.

Ch. 5: Virtual Field Trip

- Created a semi-immersive virtual field trip by geometrically, photometrically, and colorimetrically calibrating a high-resolution multi-projector display system.
- Designed an interactive field trip website on top of the Google Earth API using GPS location information and interactive versions of the panoramas.

Chapter 2

Related Work

This research covers a variety of topics and in what follows we provide a general overview of the related work. First we discuss mobile eye-tracking and the portable systems we used in the field. Next we present a review of image feature detection and matching and how it can be used to track features through a video sequence. We then discuss how image registration can be computed with these detected features using a specific image transformation model. Finally, we review techniques for calibrating multiple projector display systems.

2.1 Mobile Eye-tracking

The first eye-tracking systems were created in the 1800s and were based on mechanical devices that recorded eye movements by attaching a device directly to the observer's eye. A much less invasive technique was developed around 1900 which used photographic technology to record eye movements by taking pictures of the reflection of a light source directed towards the eye, and this is at its essence the same technique used in our mobile eye-tracking systems, albeit less sophisticated. There have been many eye-tracking systems designed based on this technique, most of which require the user to stay relatively still, and certainly not be walking around. Some examples are the lens and mirror systems used by Yarbus, electromagnetic coil systems, electrooculography systems, and the dual Purkinje system [3].

In order to take eye-tracking out into the field, we use a portable and robust system designed by Jason Babcock of Positive Science, who designed the first mobile prototype in 2003 while here at the Multidisciplinary Vision Research Laboratory (MVRL) [4, 5]. This system uses the pupil and corneal reflection method to track the eye of the person wearing it. The motion of the eye and a video of the scene are recorded using a piece of headgear that the observer wears. The base of this headgear is the lightweight frame of a pair of glasses with the lenses removed. It has mounted on it a small video camera that records what the observer is seeing and another small infrared camera mounted below the eye, illuminated by an infrared LED, that records and image of the eye and its reflections. The system is designed to be worn as a backpack that contains a laptop and various pieces of electronics that interface with the headgear (Figure 2.1).

Figure 2.1: Students wearing the mobile eye-tracking system.

The pupil and corneal reflection method used by this eye-tracking system works in the following manner. The infrared LED mounted next to the eye camera on the headgear provides illumination of the eye as well as a singular specular reflection off of the cornea, called the corneal reflection (also known as glint or the first Purkinje reflection). To estimate the gaze of the observer the center of the dark pupil and the center of the corneal reflection must first be determined by some feature detection algorithm. As the eye moves, the relative location of the corneal reflection to the pupil center changes systematically and can thus be used to calculate gaze based on this relationship [3]. However, when used outdoors on a sunny day the eye camera also records

the reflections of infrared light from the scene off the eye, which can make pupil and corneal reflection detection difficult.

The system must be calibrated before use to be able to determine what the pupil-to-corneal-reflection relationship is for accurate eye-tracking. This calibration is typically done by having the observer fixate one at a time on a set of known points in the scene. A transformation model can then be computed from this data that allows the relative distance measurement to be mapped to a gaze location on the scene video. In the field each observer wore a hat to help prevent sunlight from flooding the eye. Each time the eye-tracking system was worn, it was calibrated by having the observer fixate on a single point while moving their head through a '+' and 'x' pattern, pausing their motion several times in each segment. This procedure created the set of data points necessary for calibration the relative pupil-to-corneal-reflection distance to real gaze information. It is this gaze information that gets mapped to the panorama via the scene video in the registration algorithm of Ch. 4.

2.2 Image Feature Detection and Matching

Image feature detection is the foundation for many algorithms that rely on the knowledge of anchor points whose locations are both stable and accurate, such as camera calibration, pose estimation, 3D reconstruction [6], object tracking, and image registration [7–10]. An important example of local features are interest points (keypoints), which can be described by analyzing the appearance of regions of pixels surrounding an particular point location. Another possibility is to use edge information that can be described by local appearance and orientation, or linked together to form patterns of lines and curves. It is desirable in the case of matching images to have a feature detector that is robust to both geometric image transformations such as rotation, scale, and viewpoint, as well as image noise [11–14]. To be robust to these transformations means that the same feature can be reliably detected in an image both before and after going through these changes. This robustness makes it possible to match images that have similar content but were taken under different conditions (such as location, time, orientation, and hardware).

For the purposes of the video feature tracking and the video to panorama matching algorithm

in Ch. 4, we use a local keypoint feature detector called SIFT, which stands for Scale Invariant Feature Transform [15]. In particular, we use the version packaged with the VLFeat toolbox for MATLAB [16, 17]. The SIFT feature detection algorithm detects and converts regions of interest into an invariant descriptor representation that can be used to find matches with other descriptors. For robustness, it looks for features that are stable over multiple resolution scales of the image [18]. It uses a Gaussian function to create a multi-level image pyramid over each octave of the original image. It then subtracts these levels to make a set of Difference of Gaussian (DoG) images.

SIFT detects candidate features as extrema by comparing a pixel to the set of its 26 neighbors in the current level (8 neighbors), next upper level (9 neighbors), and previous lower level (9 neighbors). Once this feature is identified, the gradient at each pixel in a 16x16 region around the feature at the appropriate level is computed, weighted by the distance from the feature by another Gaussian function. This weighting helps to reduce the effect of gradients far from the center. Looking at the histogram of gradients, the peak is assigned as the dominant orientation. Next, this region is divided into 4x4 areas combining the gradients into an 8-bin histogram (relative to the dominant orientation) for each area. Computing the gradients this way is what makes the feature invariant to rotation. These 128 numbers form the raw descriptor of the feature keypoint. This set of numbers is saved as a normalized vector, subject to a maximum component value of 0.2. If any of the components are clipped, the vector is renormalized to unit length. Figure 2.2 shows an image with a representation of its SIFT features plotted. The size of each grid represents the scale at which the feature was found, and its orientation represents the dominant gradient at that location. The arrows within each subregion represent the dominant orientation of that section of the image [19].

To determine how well particular SIFT feature descriptors match one another, the difference between the two 128-element vectors is measured with a Euclidean norm. When a single descriptor is being matched to a set of other descriptors, a match is defined as the nearest neighbor in the Euclidean sense, subject to the constraint that the ratio of the distance between the descriptor and its nearest neighbor to the distance to its next closest neighbor is less than 80%. This ratio requirement helps to prevent matches due to noise and clutter in the image.

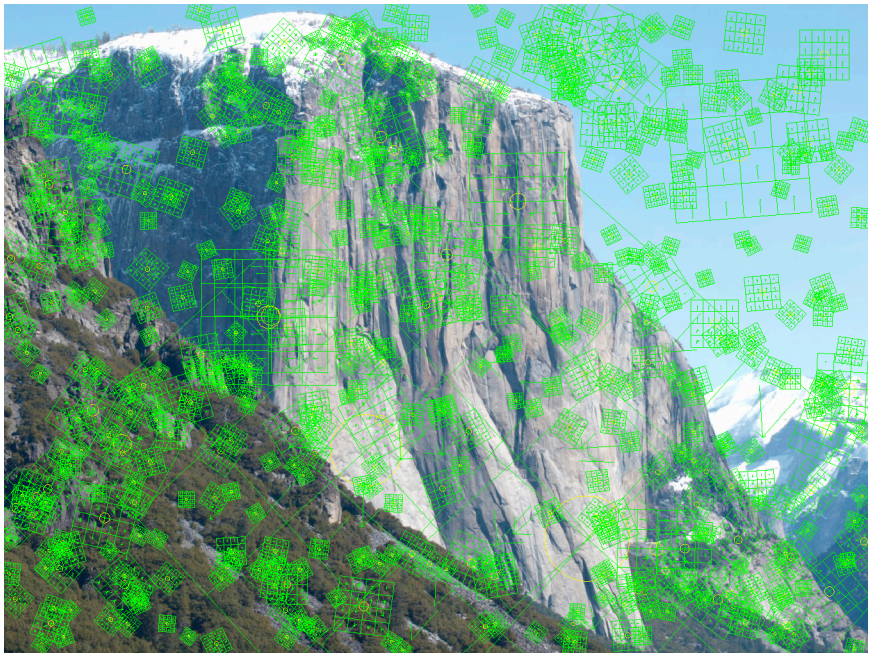


Figure 2.2: Example visualization of SIFT features detected in an image.

2.3 Image Registration Using Homographies

Image registration using local features finds applications ranging from panography and pose estimation to structure from motion and augmented reality. It is also used when creating high resolution maps by aligning and stitching together satellite imagery. When performing image registration, once the features have been detected in multiple images, a pairwise matching algorithm can be run to determine a hypothetical set of feature matches between them. This initial set of feature matches is called the putative set because it remains to be determined which are true matches and which are false positives. If a geometric transformation model is available, the matches can be fit to this model and kept only if they match the transformation sufficiently well. This can be done by selecting a small sample set to compute the transformation based on the model, then expanding that test to include the rest of the matches. This process can be repeated a number of times until a sufficiently close transformation restricted to the particular model is found, or stopped after a number of iterations if not. The process of testing a small sample and using it to verify the larger dataset is called random sample consensus (RANSAC) [20].

For our purposes, the appropriate mathematical model to use is a homography. For the

most part the panoramas we capture are of distant scenes so that the relationship between the panorama and a frame of video from the scene camera on the eye-tracker is just a pure rotation. This essentially means that the scene content we wish to match between the panorama and the video is so far away, that the change in translational position between where the panorama was captured and where the video was captured introduced a negligible amount of parallax for that region. The homography, $\boldsymbol{H}$, is an eight parameter matrix transformation model that allows matching two images with a change in position, rotation, skew, and perspective. In general, the homography transformation with point correspondances $\boldsymbol{x}_i' \leftrightarrow \boldsymbol{x}_i$ may be written as,

$$\boldsymbol{x}_i' = \boldsymbol{H}\boldsymbol{x}_i. \tag{2.1}$$

To enable a linear solution for $\boldsymbol{H}$, this equation can be rewritten in a form that makes use of the cross product,

$$\boldsymbol{x}_i' \times \boldsymbol{x}_i' = 0 = \boldsymbol{x}_i' \times \boldsymbol{H}\boldsymbol{x}_i. \tag{2.2}$$

If $\boldsymbol{x}_i' = (x_i', y_i', w_i')^\top$ and the j-th row of the matrix $\boldsymbol{H}$ is denoted by $\boldsymbol{h}^j$, this may be rewritten as,

$$\begin{bmatrix} \boldsymbol{0}^\top & -w_i'\boldsymbol{x}_i^\top & y_i'\boldsymbol{x}_i^\top \\ w_i'\boldsymbol{x}_i^\top & \boldsymbol{0}^\top & -x_i'\boldsymbol{x}_i^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{h}^{1\top} \\ \boldsymbol{h}^{2\top} \\ \boldsymbol{h}^{3\top} \end{bmatrix} = 0 = \boldsymbol{A}_i\boldsymbol{h}. \tag{2.3}$$

Since each point correspondence yields two independent equations, at least four point correspondences are necessary to find a solution for $\boldsymbol{h}$. For n matches, stack the individual $\boldsymbol{A}_i$ matrices into a single matrix $\boldsymbol{A}$. To solve for $\boldsymbol{h}$, take the singular value decomposition of $\boldsymbol{A}$,

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top. \tag{2.4}$$

The unit singular vector corresponding to the smallest singular value is the solution. If $\boldsymbol{D}$ is diagonal with positive entries arranged in descending order, then $\boldsymbol{h}$ is the last column of $\boldsymbol{V}$. This algorithm for solving for the homography is called the direct linear transformation [21]. But,

given a set of putative matches, it is necessary to perform filtration using RANSAC before trying to compute a valid homography.

RANSAC works by selecting a random sample of descriptor matches to compute an initial estimate for the particular model being used, in this case a homography, $\boldsymbol{H}$. Using this model, residuals are computed for all of the feature matches by computing the difference in the measured and mapped points:

$$r_i = \left\| \boldsymbol{x}_i' - \hat{\boldsymbol{x}}_i \right\| \tag{2.5}$$

$$\boldsymbol{x}_i' = \boldsymbol{H}\boldsymbol{x}_i \tag{2.6}$$

where r_i is the residual, $\boldsymbol{x}_i'$ are the locations mapped using the homography, and $\hat{\boldsymbol{x}}_i$ are the measured locations from the feature points. It then counts the number of inliers that are within a few pixels of their predicted location. This process repeats a for a certain number of trials until returning the sample set that yields the model with the largest number of inliers. Figure 2.3 shows an example of the result of image registration using SIFT feature descriptors and RANSAC with homography model. Before running RANSAC the putative match set has a large number of outliers. Afterwards, a group of inliers is successfully selected and an accurate homography is computed.

2.4 Projector Calibration

In general, when registering multiple projectors on a non-planar surface, multiple cameras are needed in order to reconstruct the 3D surface of the display. In [22] they use a stereo camera rig to reconstruct non-planar quadric surfaces using such techniques as conformal mapping and quadric transfer to minimize distortion after the geometric registration. Another method, [23], first achieves camera and projector calibration through 3D fiducials and then reconstructs the surface of the display by using many structured-light patterns. There are other registration methods that make use of only one camera for non-planar surfaces, however these are typically camera-view dependent. For example, [24] registers multiple projectors with respect to the viewpoint of

the camera and avoids reconstructing the display geometry entirely.

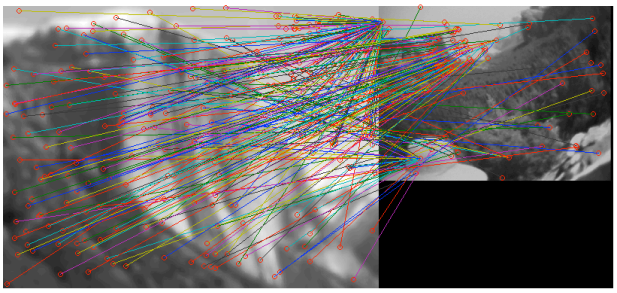
Intra-projector variation, inter-projector variation, and overlap variation are the three main categories of spatial color changes in multiple projector displays. Most existing blending methods don't address all of these issues and thus have suboptimal solutions. One of these methods, [25], proposes a gamut matching method for tiled display walls, however gamut matching significantly degrades the color quality of the display by restricting the common achievable gamut [24]. Another method, [26], matches the luminance transfer functions to achieve a luminance balancing without considering the projector chrominance variations.

In [27], they present a technique to calibrate multiple casually aligned projectors on a cylindrical surface using a single camera, where the cylinder is a vertically extruded surface and the aspect ratio of the rectangle formed by the corners of the screen is known. They achieve accurate geometric calibration of multiple projectors on a cylindrical display without performing an extensive stereo reconstruction. Our method, based on [27], is likewise able to recreate the 3D surface of the display using a single camera without needing to restrict the final viewpoint to that of the cameras position. The constrained gamut morphing algorithm created by [28] removes variations due to differences in chromaticity gamuts across the projectors, the vignetting effect of each projector, and the overlap across adjacent projectors. They demonstrate color seamlessness across multiple projectors for both planar and curved displays.

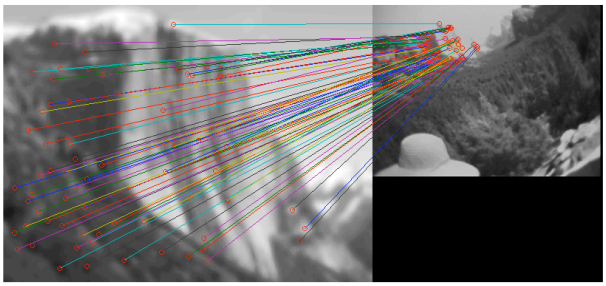
Basing our approaches on [28], the colorimeteric & photometric calibration steps we implemented consider both spatial variations in luminance within the projectors and differences in chromaticity between the projectors. By combining techniques from [27] and [28], we created a complete system for seamlessly presenting content across multiple overlapping projectors on a cylindrical surface.

(a) Scene video frame

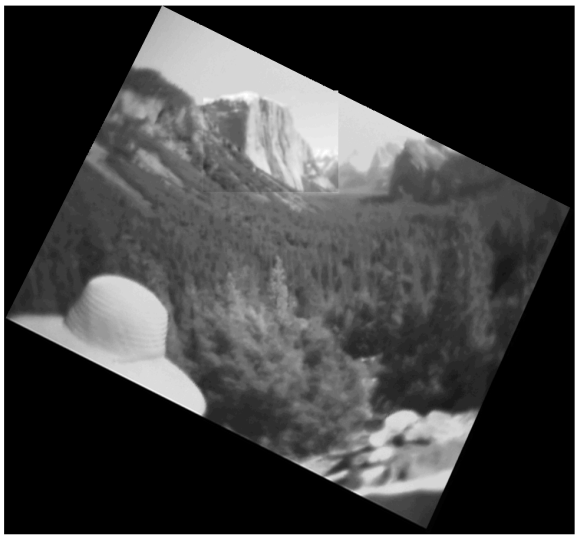
(b) Photo of El Capitan



(c) Putative SIFT feature matches



(d) Filtered SIFT feature inliers



(e) Video frame registered to photo of El Capitan

Figure 2.3: Example of image registration using SIFT and RANSAC with a homography model.

Chapter 3

Imaging in the Field

One of the main imaging goals during the field trip is to be able to capture a detailed visual representation of each important geologic site. This serves several purposes, of which an important one is being able to recreate the experience as accurately as possible within a virtual environment, while another is to use this data to aid in the eye-tracking analysis. It would also be useful to the geology education community to be able to utilize the data in a practical way for pedagogical purposes. Deciding on what type of imaging system to use depends both on these desired results as well as the practicality of using the system while in the field.

It is necessarily the case that being on the field trip involves driving (and sometime walking) to remote locations that don't have a readily available power source, visiting a potentially high number of locations during the period of a day, and facing a wide range of weather conditions over the entire trip. Thus, the imaging system must be portable and have its own power source, and because of the number of sites visited each day it must also assemble, function, and disassemble quickly and easily. The components that make up the system also need to operate well in both hot and cold temperatures and be somewhat resistant to moisture and dust.

A simple imaging system that seems to satisfy the above conditions is to use a high resolution Digital Single Lens Reflex (DSLR) camera mounted on a tripod to take an image of each site. However, the field of view for a camera with a typical lens that has a full-frame equivalent focal length of 35mm is only about 60˚ along the diagonal of the sensor. Because the important scene

elements usually cover a larger field of view it would be necessary to take multiple images to cover them. If the images were taken by rotating the camera on the tripod head they could be processed into a large panoramic image of the scene. It turns out that there are robotic systems for automating this image taking procedure which can yield enough data to create panoramas of up to several gigapixels in resolution.

3.1 Robotic Imaging Systems

3.1.1 Gigapan

The first robotic imaging system built for panoramic imaging on the field trip was based on the GigaPan EPIC 100 designed by researchers at Carnegie Mellon University in collaboration with NASA Ames Intelligent Systems Division's robotics group [29]. The GigaPan unit itself can be easily mounted on a tripod and consists mainly of a camera mounting bracket, two stepper motors run by a micro-controller and the built in software, a servo motor to physically press the camera's shutter button, and a six AA battery power supply brick. The camera chosen to be used with the GigaPan was an Olympus E-30 DSLR with an 18-60 mm zoom lens and a 12 megapixel CCD sensor. The two stepper motors control the horizontal and vertical rotations of the camera calculated by the software for the images needed to capture the panorama. The stock configuration of the GigaPan, however, needs a few modifications to make it more robust before it is ready to be used in the field.

3.1.1.1 Modifications

Instead of using the built in servo to press the camera's shutter button it would be more effective to trigger a remote shutter release via a cable plugged into the camera. Otherwise the servo tends to shake the camera while taking a picture, it draws an unnecessary amount of power, and it physically limits the placement of the camera. The EPIC 100 has a setting to enable an electronic shutter release over its remote port instead of sending the signal to control the servo. In order to use this feature, the servo must be unplugged and removed and a cable connected from the now

empty remote port of the GigaPan to the camera.

This cable must be built and wired specifically for the model of camera being used to comply with the required shutter release signal. For example when using a Hitec Servo cable to interface with the GigaPan remote port in conjunction with an Olympus camera multi connector to interface with the camera, the Olympus camera multi connector requires that both the "shutter full press" (white) and "shutter half press" (yellow) wires be connected to the yellow Hitec wire and the "shutter ground" (red) must be connected to the black Hitec wire (Figure 3.1). With this con-

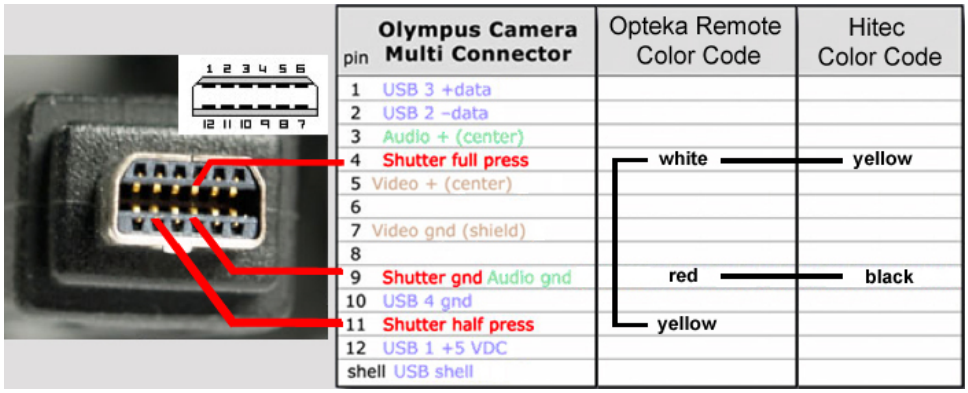


Figure 3.1: Olympus camera electronic shutter release cable wiring diagram.

figuration, the camera shutter will fire automatically when the signal is sent through the remote port without having any mechanical interactions.

The stock configuration also has some problems with handling heavier cameras like a DSLR with a relatively heavy lens attached. One reason is because all of the weight is supported on one side of the unit where the camera mounting bracket is attached to the motor while the other side of the bracket is left floating freely. Another is that the center of mass of the camera isn't centered on the rotational axis of the vertical servo motor which causes the motor to work harder to rotate the camera. To remedy this issue a support bracket was custom installed onto the side of the unit to support the camera mounting bracket (Figure 3.2). The slots on the camera mounting plate were also extended so that the camera could be mounted further back to be able balance the camera about the vertical rotation axis so that very little torque is required to rotate it up and down (Figure 3.3).

Creating a high capacity power supply to replace the standard battery pack is necessary for

(a) Front view. (b) Side view.

Figure 3.2: GigaPan unit with support bracket attached.

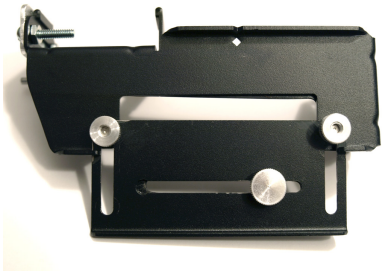


Figure 3.3: Extended camera mounting plate.

powering the GigaPan through shooting hundreds of images without having to replace the batteries too often. A battery pack that used D batteries was built for this purpose (Figure 3.4). Since it was too large to be attached to the GigaPan itself, it was mounted to the leg of the tripod and power was routed to the unit through a 2 meter long power cable terminated with "Anderson Powerpole" connectors to facilitate connecting and disconnecting the battery pack. The length of the power cable allows the GigaPan the freedom to make 360˚ rotations without getting tangled in the tripod. With this power supply the GigaPan can be used for long periods of time without having to worry about losing power during image capture and in practice lasted through several days of use during the field trip.

Figure 3.4: Portable D-cell battery pack for the GigaPan.

3.1.1.2 Panorama Capture Setup

To prepare the imaging system to capture a panorama requires assembling the components, setting the appropriate camera parameters, and configuring the GigaPan software. First, the camera is attached to the mounting bracket of the GigaPan and the remote shutter trigger cable is plugged in (after making sure the battery pack is charged and enough space is left on internal storage to hold the images for the panorama). Then the GigaPan is clipped into the tripod head at which point the battery pack can be plugged into it. This system is carried to the desired location for the point of view of the panorama, which is usually close to where the students viewing the scene will be standing. Once the tripod is setup and level, the camera parameters can be configured.

The camera should have a fixed focus, focal length, white balance and exposure throughout the entire image acquisition process to produce images that will blend together well to form the final panorama. The aperture should be set small enough to have a large depth of field so that both distant features and close foreground appear in focus; in practice it was set to around f/12. For a consistent and easy to set focal length the lens was set to its maximum value of 60 mm. This level of zoom allows for detailed visual information to be captured and generation of high-resolution panoramas. Now the GigaPan software needs to be given instructions on what type of panorama is desired so it can compute and execute the appropriate shooting sequence.

The first time a panorama is captured, the GigaPan needs to know the field of view of the camera, the amount of time to wait while the camera takes a picture and the delay necessary

after moving the camera to the next position for it to stabilize and signal it to take another. The GigaPan software calculates the field of view by having the user position the camera at two positions differing only by vertical rotation. The scene content at the top of the second position must overlap slightly with the scene content that was present at the bottom of the first position for an accurate estimate. After the field of view is set, it doesn't need to change when using the same camera and lens setup for each panorama. For the purposes of imaging in the field, the scenes were always bright enough to have a fast enough shutter speed so that the other delays could be set to their minimum values.

After the initial setup, the panorama is taken using one of two methods. The first method is to set the boundaries on the panorama by telling the software where the upper left and lower right corners should be. The second method is to tell the software to take a 360° panorama and define the top and bottom of the vertical extent of the panorama. The shooting pattern necessary for taking the panorama is then calculated and executed.

3.1.1.3 Shooting Pattern

The shooting pattern is defined by a rectangular grid with a certain number of rows and columns to zig zag through to take the images. The number of columns is fixed for all rows based on the amount needed to image at the horizon position with a minimum of 10% overlap between the images. This shooting pattern can be rather inefficient because, for example, when the camera is pointing in a direction above or below the horizon the amount of images needed is fewer than the amount at the horizon. This can lead to situations where the camera will be imaging the zenith position many times over (equivalent to the number of columns) when in reality only one image is needed. Though the shooting pattern is simple, it requires extra time to complete that wouldn't otherwise be needed. It was decided that for the following years field trip a new robotic system would be used that allowed more control over this aspect of the panorama imaging process in the hopes of being more efficient.

3.1.2 Panogear

The next generation robotic imaging system was built based off of a design of the motorized panoramic head called "Panogear" by Kolor [30]. We used a "Merlin-Orion TeleTrack GoTo" altazimuth telescope mount attached to a tripod for the main body of the system. The original purpose of this mount is to allow fine-controlled motorized tracking of celestial objects with a small telescope, however this basic functionality can be taken advantage of to provide precise control over the orientation of a camera instead. The device comes with an "Orion GoTo" computerized hand controller, a telescope mounting bracket, and a 12 volt power supply. For our purposes, a bluetooth dongle was purchased to replace the hand controller. This dongle plugs into the data port on the Merlin and wirelessly communicates with a computer running control software called "Papywizard" [31]. This software is designed for taking panoramas with the Merlin and is able to instruct it to point in a particular direction and trigger the camera to take an image. The physical trigger is sent via a cable connected from the "snap" port on the Merlin to the cameras control port. For this system the camera was upgraded to an Olympus E-PL2 DSLR with a 14-42mm zoom lens fitted with an ultraviolet filter. Though it was custom built, a couple modifications were still necessary to make it more suitable for our needs.

3.1.2.1 Modifications

A metal mounting plate for the camera was fabricated in order to be able to attach the camera to the Merlin at a vertical orientation and in such way that the principal point of the lens is centered on the horizontal and vertical rotation axes. Doing so eliminates nearly all parallax effects between photos taken while pointing at different locations in the scene and makes it easier to create an accurate panorama from the constituent photos. Additionally, as was done with the Gigapan, an external D-cell battery pack was built and attached to the leg of the tripod. This power supply allowed the unit to be portable and was able to operate the Merlin throughout the duration of the field trip on only two sets of batteries.

3.1.2.2 Panorama Capture Setup

This system requires a quick setup before its ready to take a panorama. Assuming the Merlin has been mounted to the tripod, the camera mounted to the Merlin with the snap cable connected, and the power switched on, the first thing to do is to make the connection to the bluetooth hardware dongle from the computer running the Papywizard software. Once the connection has been established, like with the Gigapan, the camera must have set a fixed focus, focal length, white balance, and exposure. For ease of use and consistency a focal length of 42mm for the lens was used. To take the panorama a predefined shooting sequence (discussed below) is selected in the Papywizard software and the panorama capture is started. During the sequence the camera captured both RAW and JPEG images. The JPEG images can be used for faster processing in the subsequent panorama stitching procedure, while the RAW images capture 12-bit data and may be used to perform nondestructive adjustments of the photos. Figure 3.5 shows a typical setup of this robotic system.

Figure 3.5: Second generation robotic imaging system.

3.1.2.3 Shooting Pattern

To capture the panorama, a sequence of overlapping images was taken by instructing the telescope mount to follow a shooting pattern specified to cover a spherical view of scene. The Papywizard software will read in an XML file specifying the order of shooting based on pitch and yaw angles. We developed an XML generator that creates this file using parameters such as the camera focal length and desired percentage overlap of images as input. The specific shooting pattern is designed to take advantage of the spherical geometry of the image capture. If a standard grid pattern is selected, an excessive amount of images are taken at any vertical angle other than the horizon. For example, if the camera is pointing away from the horizon its field of view captures a greater portion of the scene relative to the horizontal rotation axis than it does when level. Thus, we designed our pattern so that the minimum number of images was taken to guarantee coverage and ensure the appropriate amount of image overlap (Figure 3.6).

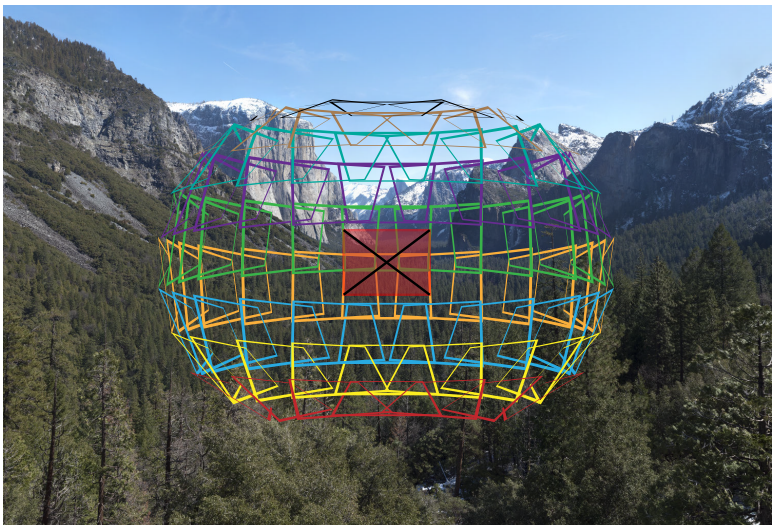


Figure 3.6: Optimized panorama shooting pattern.

The shooting pattern is also specifically designed to shoot entire rows at a time alternating about the horizon. For example, the horizon row would be captured first, then the row below the horizon, then the row above the horizon, and so on alternating back and forth until either the capture was finished or terminated. The reasons for doing this were motivated by wanting to capture the most important parts of the scene as quickly as possible. Then if, for any reason, the capture needed to be terminated early it would most likely have already gotten the desired

imagery. In this way the sites were captured more quickly and efficiently and less prone to error due to technical glitches in the hardware or software. The final set of images usually looks similar to what is shown in Figure 3.7.

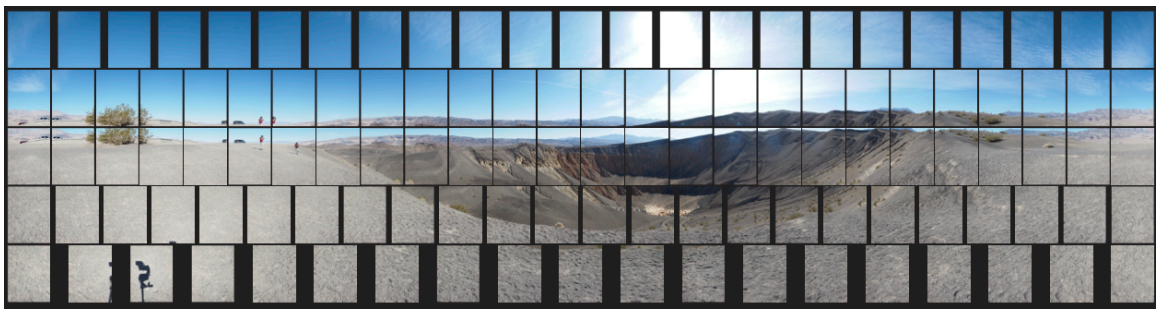


Figure 3.7: Typical results from a successful panorama capture. This particular example is of Ubehebe Crater in California.

3.1.3 Stereo Imaging

Typically, multiple panoramas of the same scene were captured simultaneously using two identical robotic systems placed a wide distance apart (Figure 3.8). The distance was measured each time

Figure 3.8: Wide-baseline stereo capture of Monte Bello Ridge in California. Notice the tripod in the distance on the left and in the foreground on the right.

with a long measuring tape to within a few inches to provide one point ground truth information. There are several important reasons for doing this. Firstly, it provided a failsafe mechanism in

case one of the devices malfunctioned and prevented capture of a scene. Secondly, it provides an option to choose which panorama is best suited for presentation in the virtual environment and that has best captured all of the features of geologic importance. On this note, one of the tripods was placed very close to the perspective that the students would have while viewing the scene. This makes the task of mapping gaze data to the panoramas easier as well. Finally, it opens up the possibility of generating some sparse 3D geometry about the scene from the widely separated pairs of images.

3.2 Imaging Augmented with GPS

Two identical hand-held Garmin GPS units, an "active" unit and a "base" unit, were used to measure various pieces of location information particularly useful for the field trip. The base unit was run continuously throughout the day in the vehicle taking a measurement every second to keep a record of where we went, when, and for how long. To increase the chances of an accurate signal an external antenna was mounted to the roof of the vehicle and routed in through the driver side door to connect to the hand-held unit. Digital markers were set every time we arrived and left a stop to make it easier to analyze the location information afterwards.

To augment our panoramas with ground truth location data, the active unit was used to keep track of the position of each tripod as it was placed in the field at each site. This also provided a second measurement of the distance between tripods when stereo panoramas were being captured. The active unit was carried around at the site while taking individual photos of the imaging setup and scene to provide more reference frames for analyzing the geometry of the scene. The times on the cameras were manually synchronized to the GPS time so that the locations of the photographs could be determined afterwards without having to manually mark them.

Because the GPS measurements have some error associated with them it would be desired to have a way to reduce the error by combining the measurements from the active and base units. By calculating the fluctuations away from average in the base unit might be possible to compensate for any location data fluctuations in the active unit, yielding more accurate location information for all data recorded at a site.

3.3 Scene Video from Eye-trackers

In addition to the high-resolution image capture systems, there were video cameras attached to the headgear of every eye-tracker to capture a view of what each person was observing. They are typically used for calibrating and visualizing the gaze of each individual observer onto the video of the scene (Figure 3.9), but also provide another source of content that may eventually be incorporated into the virtual field trip. The fact that there are a large number of scene videos

Figure 3.9: A sample video frame from the eye-tracker scene camera, overlaid with the eye camera and gaze data.

from slightly different viewpoints provides interesting possibilities for tracking the pose of each observer and simultaneously visualizing the gaze of each observer in a common 3D reference frame. The scene videos are low resolution and have poor color fidelity, but have a wider than typical field of view because of the use of a wide-angle lens. Unfortunately, because of this lens, there is a some amount of lens distortion present in the video which should be compensated for before doing any image analysis. One way of doing this is to image a known object, like a checkerboard, and calculate the radial distortion model necessary to straighten out all of the curved lines [32]. Due to the nature of the wide dynamic range that comes from being outdoors and the use of automatic gain control, it is impossible to capture all of the detail that exists in the highlights and shadows of the scene at once. These particular shortcomings make it difficult to

perform comparisons between the video content and the ultra high fidelity panoramas. A portion of the next chapter discusses a method for mapping observer gaze onto the panorama using their video as a guide despite this difficulty; a step towards realizing the goal of a common reference frame.

Chapter 4

Image Processing

4.1 Panorama Generation

Generating the high-resolution panoramas requires an image stitching and blending procedure to combine all of the individual photographs from a particular capture into a unified representation of the scene. Many software packages are available for this procedure, but we have found the Autopano Giga program by Kolor to be the fastest and most reliable for our needs [33]. This software detects and builds panoramas from an arbitrary set of input images. It uses SIFT feature detection and matching to fit a similarity or homography image transformation model in conjunction with outlier rejection to generate the final image feature correspondences. If desired, there is an option to simultaneously correct for any lens distortions that may be present in the images.

The Autopano Giga software also has the ability to take into account the particular shooting pattern used with either the GigaPan robot or PapyWizard software when making its calculations. For images taken with the GigaPan one may select the shooting pattern that corresponds with what was setup in the GigaPan software. For images taken using the Papywizard software, the XML file that specifies the shooting pattern can be selected. In both cases a small preview is generated showing the input images in the locations determined by the shooting pattern. This way any positioning errors can be spotted before starting the image matching procedure. Using

these import options typically produces a more accurately stitched panorama. Once this stitching step is completed, the panorama is visualized using one of several projection methods prior to the final rendering.

4.1.1 Projection Methods

When visualizing a spherically captured panorama, it is typical to use some projection method to represent the entire scene on a two dimensional image plane – analogous to creating a map of the earth. Autopano Giga has several projection options, namely spherical, cylindrical, and planar. Also known as equirectangular projection, spherical projection is simply an angular parameterization of the panorama. In a panorama h pixels tall and w pixels wide, the x and y coordinates are related to the longitude, θ, and the latitude, ϕ, by,

$$\begin{aligned} x &= \frac{w}{2}\left(1+\frac{\theta}{\pi}\right) \\ y &= \frac{h}{2}\left(1-\frac{2\phi}{\pi}\right) \end{aligned} \tag{4.1}$$

where $-\pi \leq \theta \leq \pi$ and $-\pi/2 \leq \phi \leq \pi/2$. In the resulting panoramic image, the vertical axis represents the latitude and the horizontal axis represents the longitude. This is the rendering method most easily used for creating interactive versions of the panorama in which the viewer looks around the scene mapped onto a sphere, as if they were themselves in the location at which the panorama was taken. Figure 4.1 shows an equirectangular rendering of the set of images shown previously in Figure 3.7. The black areas above and below the image content represent regions of the scene that weren't captured.

To compute a cylindrical projection, ray trace from the center of projection through a cylinder centered about that point out until an intersection with one or more images. Then, color that particular pixel on the cylinder with the color determined by the rendering algorithm for the pixels from original images that lie along this ray (Figure 4.2). For a cylindrical projection with a vertical field of view, $V_{FOV} < \pi$, the x and y coordinates are related to the longitude, θ, and the

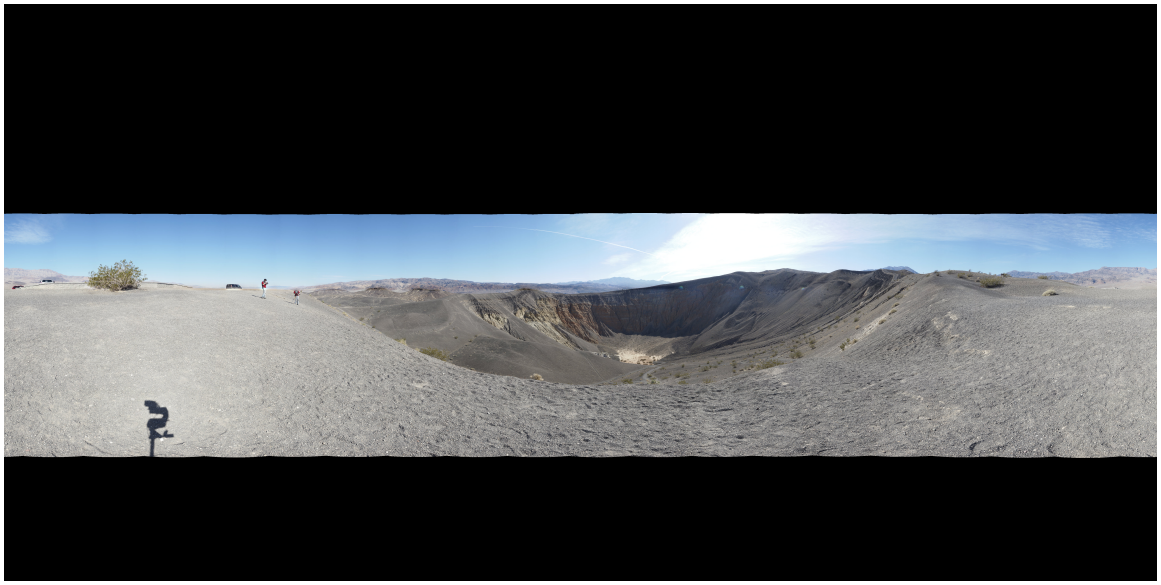


Figure 4.1: Equirectangular (spherical) rendering of the panorama taken at the Ubehebe Crater in California. The black areas represent portions of the sphere where image data was not captured.

latitude, ϕ, by,

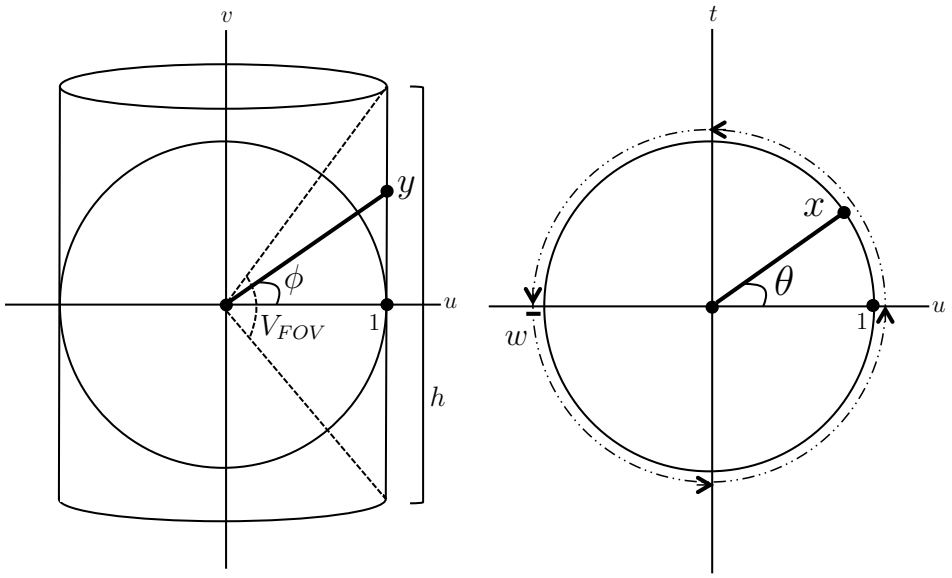
$$\begin{aligned} x &= \frac{w}{2}\left(1+\frac{\theta}{\pi}\right) \\ y &= \frac{h}{2}\left(1-\frac{\tan\phi}{\tan\frac{V_{FOV}}{2}}\right) \end{aligned} \tag{4.2}$$

where $-\pi \leq \theta \leq \pi$ and $-V_{FOV}/2 \leq \phi \leq V_{FOV}/2$.

A similar procedure is used for planar projection, replacing the cylinder with a plane generating a particular field of view placed about the center of projection (Figure 4.3). For a planar projection with a vertical field of view, $V_{FOV} < \pi$ and a horizontal field of view $H_{FOV} < \pi$ the x and y coordinates are related to the longitude, θ, and the latitude, ϕ, by,

$$\begin{aligned} x &= \frac{w}{2}\left(1+\frac{\tan\theta}{\tan\frac{H_{FOV}}{2}}\right) \\ y &= \frac{h}{2}\left(1-\frac{\tan\phi}{\tan\frac{V_{FOV}}{2}}\right) \end{aligned} \tag{4.3}$$

where $-H_{FOV}/2 \leq \phi \leq H_{FOV}/2$ and $-V_{FOV}/2 \leq \phi \leq V_{FOV}/2$.



(a) Front view.

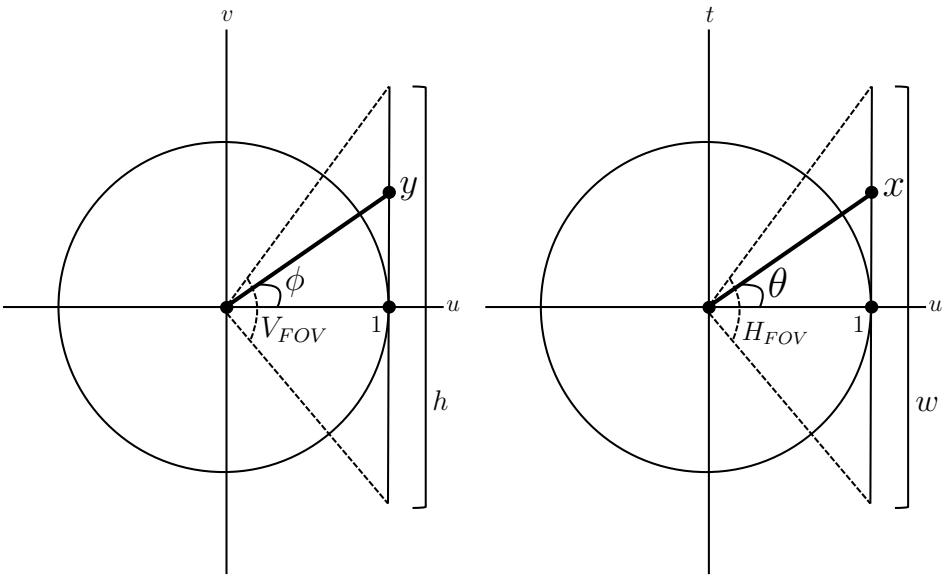
(b) Top view.

Figure 4.2: Spherical to Cylindrical Panorama Projection Diagram

Any of these methods naturally introduces some amount of geometric distortion. The trade off is between generating a physically accurate representation of the scene according to the human visual system and creating a field of view large enough to be representative of the geology in the scene.

4.1.2 Rendering Methods

After choosing the desired projection method, the panorama can be rendered to an image file. Autopano Giga has a smart blending algorithm built in that automatically blends overlapping photos from the panorama. The algorithm detects and removes ghosting and motion artifacts and guarantees a smooth transition between any exposure or color change between images. The algorithm is multi-threaded and GPU accelerated which speeds up the process significantly. If exported at full resolution, it is possible to create panoramas with resolutions of several gigapixels. The panorama is now ready to be used for use in video registration and the virtual environment.



(a) Front view. (b) Top view.

Figure 4.3: Spherical to Planar Panorama Projection Diagram

4.2 Video to Panorama Registration

The purpose of registering the scene video to the high resolution panorama is to be able to have a common reference frame for analysis of the eye-tracking data. What this means is that instead of analyzing gaze data on an individual basis, the gaze data from a group of observers can be seen and analyzed simultaneously on a single reference image. The difficulty in doing this is in part because every scene video and the panorama itself are captured from a different point of view relative to the scene and each other, and in part because the image quality of the scene videos is vastly inferior to the image quality of the panorama.

A baseline method for accomplishing this task relies on the mathematical matrix transformation known as a homography. It is a linear projective mapping that takes points from one plane and maps them to another. When applied to images and video taken of a scene, it can transfer points from one image to another via an intermediate scene plane. Thus, each video frame should have a homography mapping it to the panorama. It works under the assumption that the images have no nonlinear distortions and are perfect planar projections. In our case the eye-tracker scene videos have lens distortion that we choose to ignore, the panorama has been rendered us-

ing a spherical projection, and there usually isn't a single common scene plane present to identify feature points with. Additionally, the different viewpoints create problems with occluded scene information. However, in many outdoor scenes the features of importance are far away and the parallax between a given scene video and the panorama is negligible in that region. In this case it is the plane at "infinity" that acts as the intermediate scene plane. Though even if this isn't the case, the matching process can still be attempted.

4.2.1 SIFT Feature Tracking

Before matching video frames to the panorama, a scene video is analyzed to determine the frame to frame image transformations. This allows for the calculation of temporally consistent and smooth results when matching individual video frames to the panorama. To do this, SIFT features are computed and matched between each frame and its neighbor using the VLFeat toolbox. Generally there are a large number of outliers after matching and they need to be filtered out according to a specific image transformation model. Since the change in baseline frame to frame in a video is small with a negligible amount of parallax, a homography model can be used to approximate the transformation from one frame to the next. Using functions from Peter Kovesi's computer vision and image processing toolbox [34], the putative SIFT matches are passed to a RANSAC procedure that uses this model to determine a set of matches which are inliers to the model. It is these matches that are used to compute the homography. A record is kept of the inlier matches from frame to frame so that feature tracks can be determined during processing.

The videos and initial SIFT data are put through a second-pass version of this method that uses only the SIFT features that last through a minimum specified number of frames to recompute the frame-to-frame homography transformations. This method promotes the use of robust SIFT features and ignores those noisy features which disappear quickly. Figure 4.4 graphically demonstrates SIFT tracking performed on a particular video frame. The different colors represent how many frames the tracks have survived: green is greater than 15, yellow is between 7 and 14, and red is less than 7.

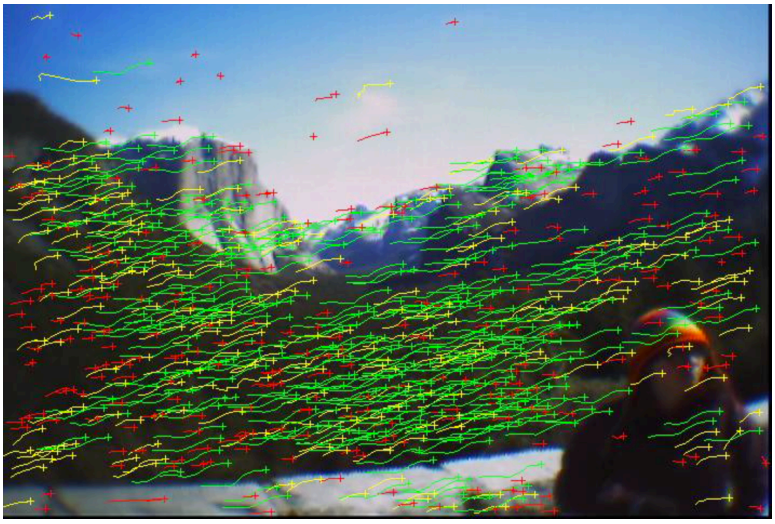


Figure 4.4: A visualization of SIFT feature tracking on a video frame. The different colors represent how many frames the tracks have survived.

4.2.2 Keyframe Selection

Now that the frame-to-frame homographies have been computed for the scene video, video to panorama registration can be attempted. Each frame of the scene video needs to be related to the panorama by some kind of spatial transformation. As discussed above, a homography model can be used to accomplish this. One way to do this for the entire video sequence is to select a single keyframe from the video that will act as a homography anchor to the panorama. Let the keyframe be given by the kth video frame and let the current video frame be the jth. To compute the appropriate homography for every other video frame, compose each frame-to-frame homography between the given frame and the keyframe followed by applying the homography from the keyframe to the panorama, $\boldsymbol{H}_{k,p}$. Then (if $j < k$),

$$\begin{aligned} \boldsymbol{H}_{j,k} &= \prod_{l=j}^{k-1} \boldsymbol{H}_{l,l+1} \\ \boldsymbol{H}_{j,p} &= \boldsymbol{H}_{k,p}\boldsymbol{H}_{j,k}. \end{aligned} \tag{4.4}$$

A similar procedure works for when $j > k$. The first computation will yield a homography relating the given frame to the keyframe, $\boldsymbol{H}_{j,k}$, and the second computation will upgrade that transformation to a relationship between the given frame and the panorama, $\boldsymbol{H}_{j,p}$. This method

ensures that the smoothness of the frame-to-frame transformations is preserved when mapping each video frame to the panorama. The drawback, however, is that the further a given frame is from the keyframe that anchors the video to the panorama, the less accurate the transformation becomes. This phenomenon is called drift (because the solution is drifting away from what is desired) and is mainly caused by accumulations of very small errors in the frame-to-frame homographies summed over a large number of video frames. The errors themselves are due simply to imperfect data and noise when analyzing the video frames during the SIFT feature tracking procedure.

To counteract drift in the solution, multiple video keyframes can be selected to anchor the spatial transformation between the scene video and the panorama. This serves to "reset" the drift that has accumulated at each keyframe, yielding more accurate results. To implement this idea, the keyframes need to be selected in such a way that they are distributed fairly well throughout the entire video sequence. One way to do this is to simply select a keyframe every certain number of frames so that there is an anchor at equal time interval during the video.

A more sophisticated method is to use the results of the single keyframe method to inform the algorithm where to place additional keyframes. By looking at the projective components of the homographies relating each video frame to the panorama, it is possible to determine which frame in a given time interval is closest to an affine transformation and set it as a keyframe. This is done by analyzing the magnitude of the projective components over time and selecting the local minima as the keyframes. To avoid getting a keyframe too often, a parameter is used to smooth the data slightly before computing the local minima. This smoothing parameter essentially controls the desired keyframe density of the result.

Once the set of keyframes has been selected, each of them needs to be matched to the panorama with a homography transformation. This is done in the same way as computing the frame-to-frame transformations in the video sequence, with just a few changes. First, SIFT features are computed for the panorama which will be used for every video keyframe analysis. Then, for each keyframe, SIFT features are computed and RANSAC with a homography model is run to compute feature match inliers between the keyframe and the panorama and to compute the resulting homography. It turns out that due to the large differences in image quality between the video

frames and the panorama that this procedure as it stands will typically fail.

What has been found to work more robustly is to do SIFT feature detection going through ever increasing levels per octave of computation. Thus, the first computation compares SIFT features detected at three levels per octave to see if a valid homography can be determined using RANSAC. If not, the procedure is repeated at four levels per octave all the way up until twelve levels per octave if necessary. Once a valid homography is found, the process repeats for the next keyframe. In the case that no valid homography is found, the algorithm resorts to asking for a manual selection of control points that match between the video frame and panorama. At this point the user also has the option of rejecting that particular keyframe entirely and moving onto the next one.

4.2.3 Final Homography Calculation

Once the keyframe to panorama homographies have been calculated, the frame to panorama homographies for all other video frames are computed by composing the frame-to-frame homographies with the neighboring keyframe to panorama homographies. Thus, if k_1 represents the previous keyframe, k_2 represents the next keyframe and the current frame is the jth, there will be two frame to panorama homographies given by,

$$\begin{aligned} \boldsymbol{H}_{j,k_1} &= \prod_{l=j}^{k_1+1} \boldsymbol{H}_{l,l-1} \\ \boldsymbol{H}_{j,p_1} &= \boldsymbol{H}_{k_1,p}\boldsymbol{H}_{j,k_1} \end{aligned} \tag{4.5}$$

$$\begin{aligned} \boldsymbol{H}_{j,k_2} &= \prod_{l=j}^{k_2-1} \boldsymbol{H}_{l,l+1} \\ \boldsymbol{H}_{j,p_2} &= \boldsymbol{H}_{k_2,p}\boldsymbol{H}_{j,k_2}. \end{aligned} \tag{4.6}$$

To ensure a smooth transition between keyframes and to counter any errors due to drift, the homography at each frame is computed as a linear combination of the transformations computed

using its surrounding keyframes, weighted by the distance to each keyframe,

$$\begin{aligned} \boldsymbol{H}_{j,p} &= \alpha(j)\boldsymbol{H}_{j,p_1} + \beta(j)\boldsymbol{H}_{j,p_2} \\ \alpha(j) &= \frac{k_2 - j}{k_2 - k_1} \\ \beta(j) &= 1 - \alpha(j). \end{aligned} \tag{4.7}$$

The final set of video frame to panorama homographies can then be used to visualize the scene video and eye-tracking data onto the panorama (Figure 4.5).

Figure 4.5: A visualization of a frame of the scene video mapped onto the panorama via its homography.

4.2.4 Ground Truth Evaluation

To determine how well this algorithm is performing there needs to be a way of comparing its results against a ground truth measurement. Unfortunately, accurate ground truth data isn't available so the alternative is to visually create some approximate ground truth data. To do this, nine control points representative of the important scene content were manually selected in a panorama of Inspiration Point that would be visible throughout a particular 30 second video sequence (Figure 4.6). Then, points were located manually in the video sequence that matched the panorama control points (if visible) for every ten frames.

With this new set of data and the previous set of video to panorama homography transformations, a measurement of error can be computed. This error, E, is defined geometrically as the Euclidean distance (in pixels) between the manually selected point in the jth video frame, $\boldsymbol{x}_j$, mapped via its corresponding homography, $\boldsymbol{H}_{j,p}$, to a point on the panorama, $\boldsymbol{x}'_j$, and the

Figure 4.6: The nine control points selected for use in the ground truth evaluation of the scene at Inspiration Point, Yosemite National Park, California.

appropriate control point, $\boldsymbol{p}$, in the panorama [35],

$$\begin{aligned} \boldsymbol{x}'_j &= \boldsymbol{H}_{j,p}\boldsymbol{x}_j \\ E &= \left\| \boldsymbol{p} - \boldsymbol{x}'_j \right\|. \end{aligned} \tag{4.8}$$

Ideally they will each be the same point and the error value will reduce to zero pixels. In reality,

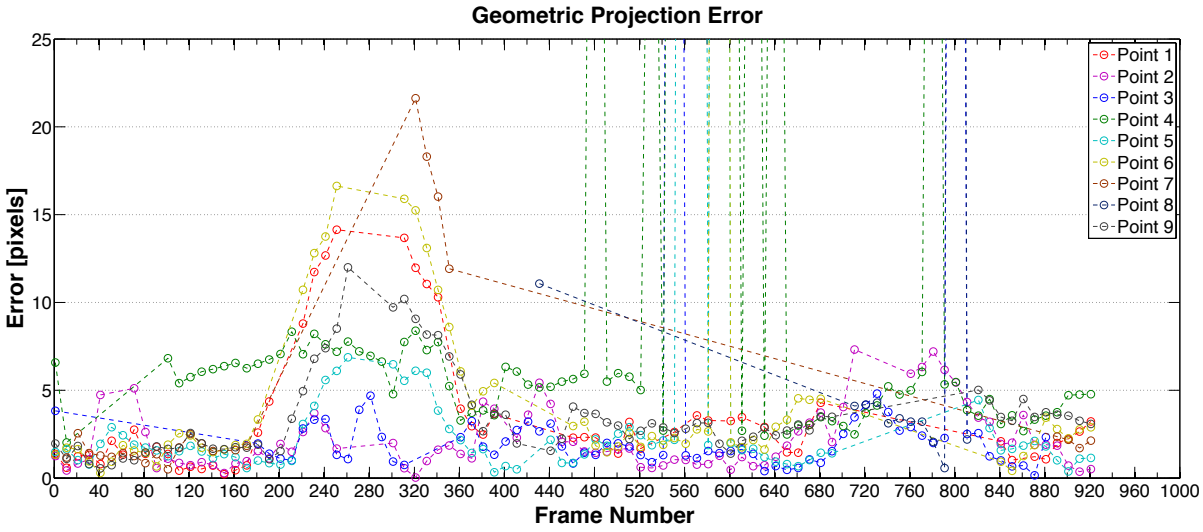


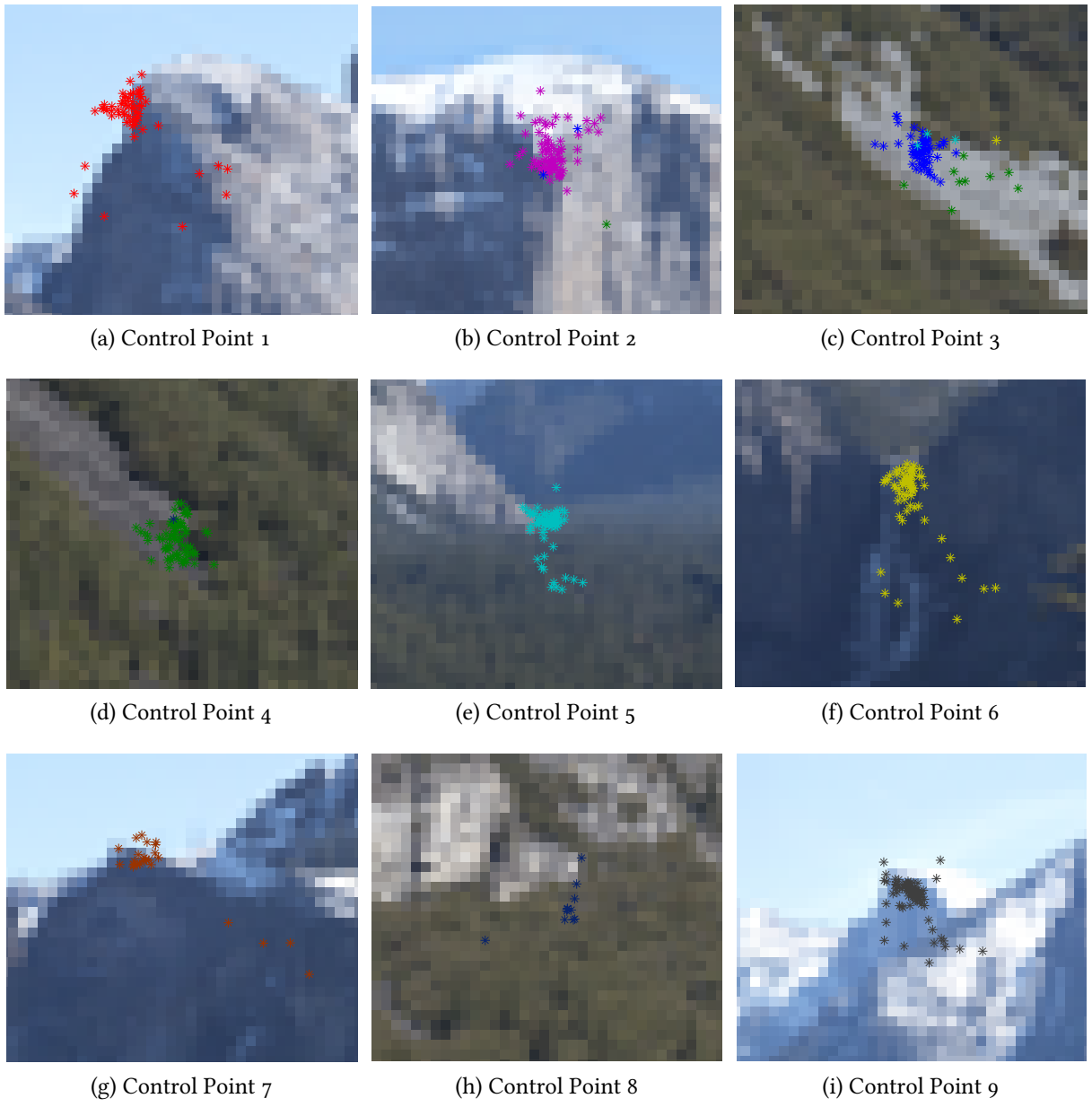
Figure 4.7: The geometric projection error in pixels on the panorama, computed from the manual points selected in each video frame corresponding to each of the nine control points.

the results are not perfect and have a variable amount of error, as plotted in Figure 4.7. The error measurements were made in the reference frame of the panorama, which has a down-sampled resolution of 2000px horizontally by 500px vertically.

Figure 4.8: Plot of the mapped video points for each of the nine control points.

Most of the mapped video points lie within a distance of about 5 pixels to their respective control point on the panorama. It turns out the points that have errors that go high off the chart are due to user mistakes when finding the control points in the video sequence. Looking at the raw data it is easy to see that a control point other than the one that was being worked on was selected, causing those enormous errors. Figure 4.8 shows the mapped video points plotted on the panorama for each of the control points and Figure 4.9 shows a close-up view of the area around each control point. Notice how some of the control points have mapped points whose colors do not match; this is due to the user error in selecting the correct control point. The rest of the data (ignoring mistakes) lie between a distance of about 5 and 25 pixels to their respective control point. Thus, a majority of the mapped points are matched well to their control point, however around each there is a scattered amount of poorly matched points (within the 25 pixel range). Figure 4.10 shows box plots of the statistical distribution of geometric projection errors corresponding to each of the nine control points. On each box, the center line represents the median, the edges of the box are the 25th and 75th percentiles, the whiskers extend to the most extreme data points not considered outliers, and supposed outliers are plotted individually.

It should be possible to improve these results further by first removing the lens distortion present in the scene videos and by using a portion of the panorama that can be rendered linearly instead of equirectangularly. Doing so would bring the data closer to the original set of assumptions for this model to work well and should provide a more accurate mapping.



(a) Control Point 1 (b) Control Point 2 (c) Control Point 3

(d) Control Point 4 (e) Control Point 5 (f) Control Point 6

(g) Control Point 7 (h) Control Point 8 (i) Control Point 9

Figure 4.9: Close-up views of the mapped video points for each of the nine control points.

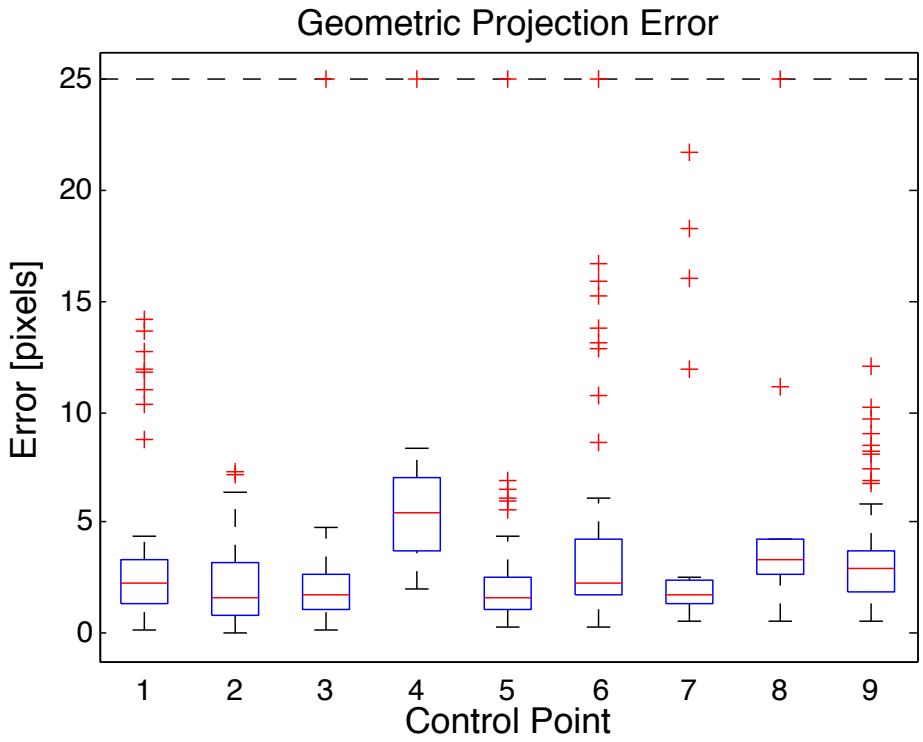


Figure 4.10: Box plots of the statistical distribution of geometric projection errors corresponding to each of the nine control points.

Chapter 5

Virtual Field Trip

5.1 Semi-immersive Virtual Environment

One of the main goals of this project is to understand how novice students learn while making observations in the field as compared to how they learn while making similar observations using a virtual environment. For this purpose, we have a large scale display system that consists of four tiled projectors and a curved widescreen display surface. This system is used to present large high resolution images that are visually representative of particular geologic sites. The space is large enough for small groups of students to observe the images with each other while they are being eye-tracked. In an ideal situation, the projector-display system is calibrated such that it is possible to present the images as if one were viewing them through a large window, meaning the system has been geometrically aligned, photometrically and colorimetrically calibrated, and the output of each projector blends seamlessly into output from adjacent projectors.

5.1.1 Multi-projector Calibration

The two main steps involved in calibrating the virtual environment display system are geometric registration, in which the geometric alignment between all projectors and the display surface is determined, and photometric and colorimetric calibration and blending between projectors which allows for the display of a seamless image across the display surface. The geometric registration

step makes it possible to render a panorama which will appear as one continuous image across the display surface, taking into account the geometry of the projection wall and the relative alignment of the four projectors. The photometric and colorimetric calibration step adjusts the panorama image portions sent to each projector, via an alpha mask, in such a way that the colors closely match across the different projectors and the projected image blends smoothly from one projector to the next for regions in which they overlap. Figure 5.1 is a flow chart depicting an overview of the major and minor steps necessary to fully calibrate this display system.

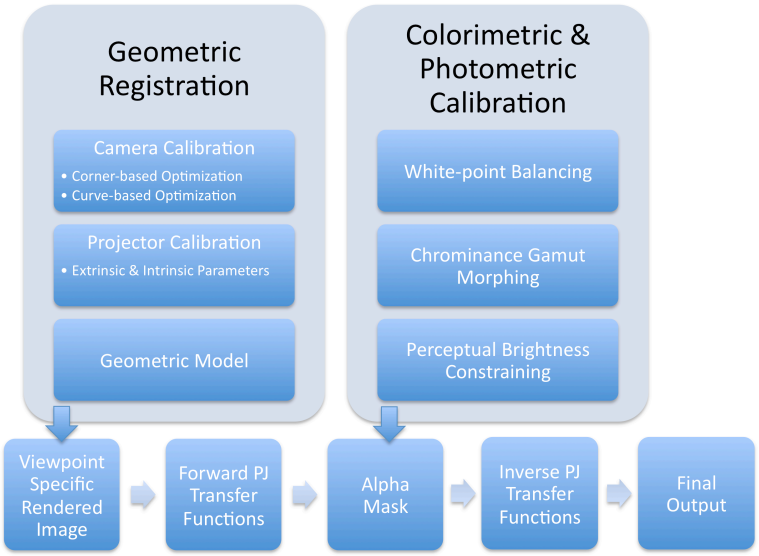


Figure 5.1: Flow chart describing the projector calibration algorithm.

5.1.1.1 Geometric Registration

Geometric registration of this projector display system is performed using the technique in [27]. This technique computes the projector to display geometric relationships through a three step process. Using a calibrated camera, a single image is taken off axis relative to the display surface of both the display itself and the specific light patterns displayed upon it by each projector (Figures 5.2 and 5.3). These light patterns determine the top and bottom contours as well as the corners of each projectors view frustum which are used in the third geometric calibration step. The first and second steps compute an accurate camera orientation relative to the display surface. First, a world coordinate system is defined such that the origin lies at the center of the virtual plane

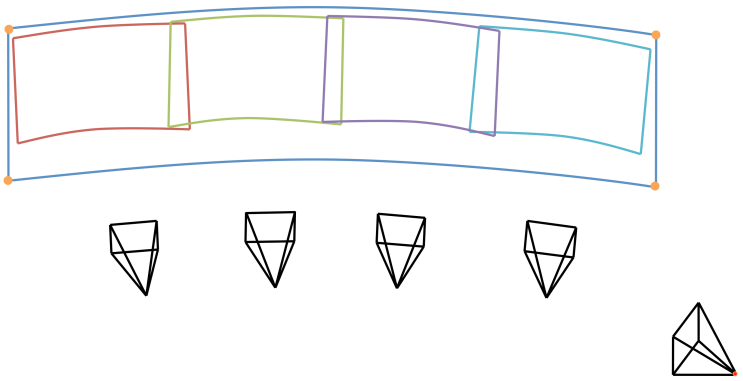


Figure 5.2: Typical arrangement of projectors, camera, and cylindrical projection surface.

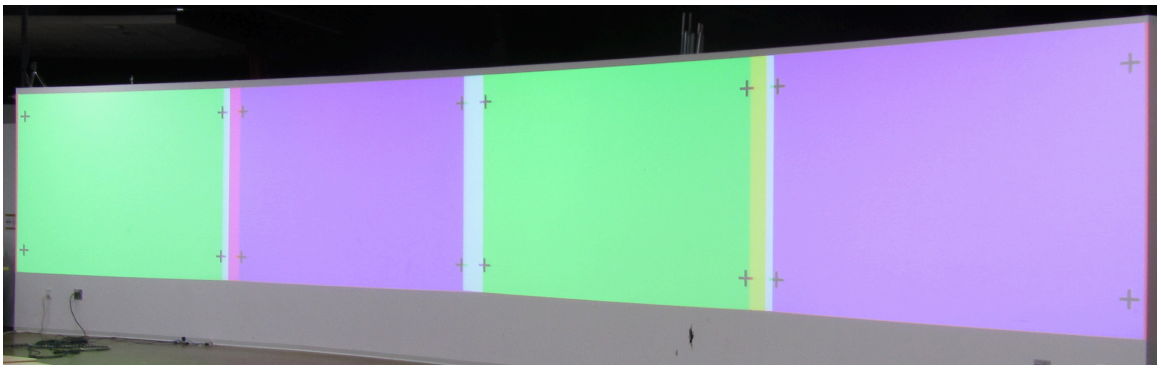


Figure 5.3: The light pattern used to geometrically calibrate the projectors and display wall.

created by the four corners of the display. The top of the display is defined as the $Y = 1$ plane and the bottom of the display is defined as $Y = -1$ (Figure 5.4).

Then the corner-based optimization step is performed. Using the imposed 3D coordinates of the corners of the display and measuring their pixel locations in the camera's image, a Levenberg-Marquardt nonlinear optimization process is used to solve for the focal length, f, and extrinsic orientation (rotation, $\boldsymbol{R}$, and translation, $\boldsymbol{T}$) of the camera in the world coordinate system by minimizing reprojection error. The reprojection error is defined as the distance between the predicted display corners, $\boldsymbol{c}_{pred}$, and projected display corners, $\boldsymbol{c}_{proj}$, given the current estimate of the camera parameters,

$$\begin{aligned} &\min_{\theta} \sum \left\| \boldsymbol{c}_{pred}(\theta) - \boldsymbol{c}_{proj} \right\|^2 \\ &\theta = [f, \boldsymbol{R}, \boldsymbol{T}]. \end{aligned} \tag{5.1}$$

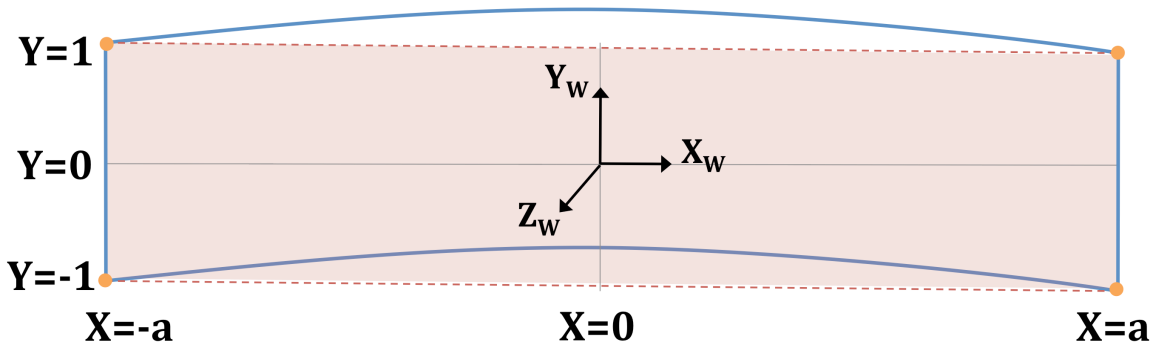


Figure 5.4: Cylindrical surface and world coordinate system.

After this first optimization step there is a refinement step which is based upon the projections of the top and bottom curves of the display into the camera's image plane. Points in the image along the top and bottom curves of the display are used to estimate a quadratic function to represent them. Using the current camera parameters as an initialization, the quadratic curve that approximates the top of the display is backprojected and intersected with the $Y = 1$ plane, which represents the top of the display. This yields a set of 3D world coordinates which represent the location of the top curve of the display. Assuming a vertically extruded surface, these points are translated to the $Y = -1$ plane (the bottom of the display) and reprojected into the camera's image plane. This curve, $\boldsymbol{x}_p$, is compared with the measured curve, $\boldsymbol{x}_b$, and once again a Levenberg-Marquardt nonlinear optimization process is used to refine the focal length and camera orientation such that these curves match as closely as possible in minimizing a least squares error measurement,

$$\min_{\theta} \sum \left\| \boldsymbol{x}_p(\theta) - \boldsymbol{x}_b \right\|^2 . \tag{5.2}$$

The geometry of the projectors themselves can now be calibrated, using the current camera calibration, the display surface geometry, and the image data taken of the projected light patterns. The horizontal axis, $\boldsymbol{X_P}$, of each projector is computed by calculating the planes formed by a set of points from the top and bottom lines of the projected light pattern and finding where they intersect. The center of projection, $\boldsymbol{O_P}$, lies along this line at the point where the planes connecting the vertical corners of the light pattern intersect to form an equal angle on either side relative to the horizontal axis. The direction of the forward looking axis, $\boldsymbol{Z_P}$, is orthogonal to the horizontal axis, constrained by the vertical plane which intersects the two horizontal planes

with equal length. Once these axes have been determined, the orthogonal projector coordinate system is completed by defining the third axis, $\boldsymbol{Y_P}$, as the cross product of the $\boldsymbol{Z_P}$ and $\boldsymbol{X_P}$ axes (Figure 5.5).

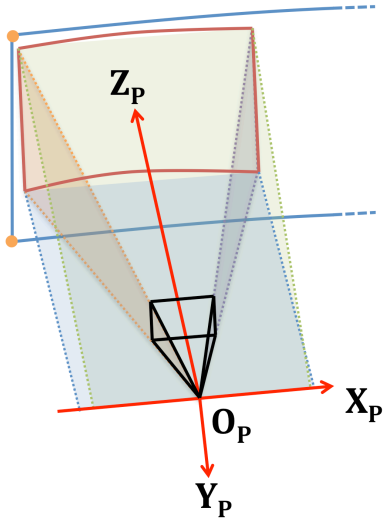


Figure 5.5: Local coordinate system for projector calibration.

5.1.1.2 Photometric and Colorimetric Calibration

The next step is to compensate for differences in luminance and color between the projectors and to seamlessly blend together the regions of overlapping light. The color primaries of each channel of each projector are measured with a colorimeter, and the two dimensional luminance profile of each projector is measured using a calibrated camera. A white-point balancing step is performed that computes a per color channel scale factor, α_l, which will change the measured color primaries, $(x_l, y_l)^\top$, of each projector to match the chromaticity coordinates of the desired white-point, $(x_D, y_D)^\top$. We solve for the scale factors using,

$$\frac{\sum_l \alpha_l B_l (x_l, y_l)^\top}{\sum_l \alpha_l B_l} = (x_D, y_D)^\top \tag{5.3}$$

and by fixing $\alpha_R = 1$, where $B_l = X_l + Y_l + Z_l$. The values of α_l are then normalized by the scale factor with the largest value. Next, using the measured luminance profiles and known areas of overlapping light, an "s-curve" is computed horizontally for each region that transitions smoothly

from one projector to the next,

$$L_D = \tau L_{E_1} + (1 - \tau) L_{E_2} \tag{5.4}$$

where L_D is the desired luminance, L_{E_1} is the luminance of the first edge, L_{E_2} is the luminance of the second edge, and τ is a value from 0-1 representing the contributions of each defined by the "s-curve". Additionally, the individual contributions from each projector to satisfy this desired luminance curve are constrained to fall off as a smooth curve to avoid any hard edges. Alpha masks are then created from the computed luminance contribution maps to be applied to the images before projection.

To refine the brightness variations leftover after the gamut morphing, a perception based gradient constraint is applied to the entire luminance map [28, 36]. There are three constraints enforced, the first of which is called the "capability" constraint. It ensures that the modified luminance, W_l', never exceeds the maximum luminance achievable by the display. The "perceptual uniformity" constraint maintains a smooth variation in the luminance by requiring

$$\bigtriangledown W_l' < \frac{1}{\lambda} W_l', \tag{5.5}$$

where λ is the perceptually based smoothing parameter [28]. Finally, the "display quality objective function" increases the dynamic range by ensuring that the integration of the luminance function over the display is maximized.

5.1.2 Rendering Images for Display

Finally, an image can be rendered for display. The process is done by first raytracing the image based on the projector display geometry previously calculated and then applying the white point compensation and edge blending (via the alpha masks) to each individually projected image segment. When displayed, the result will be a seamless image across the tiled projectors ready to be used for viewing (Figure 5.6).

5.1.3 Projector Tiling Solution

For our current implementation of the virtual environment, however, an alternative method was used. This is because the slight changes in color between projectors that still remained after calibration actually have a big effect on the geological interpretation of the scene. It was decided to instead tile the projected images next to one another as closely as possible without overlapping. Then, to mask any deficiencies in the process and make clear the distinction between projectors, a thin vertical black line was inserted into the presented imagery at the location of the seams between the projected images (Figure 5.7). Doing so made it obvious to ignore the gaps and not to interpret the seams as any type of geologic feature.

(a) Uncorrected image

(b) After geometric registration only

(c) Geometric registration with photometric & colorimetric calibration

Figure 5.6: Three stages of projector calibration.

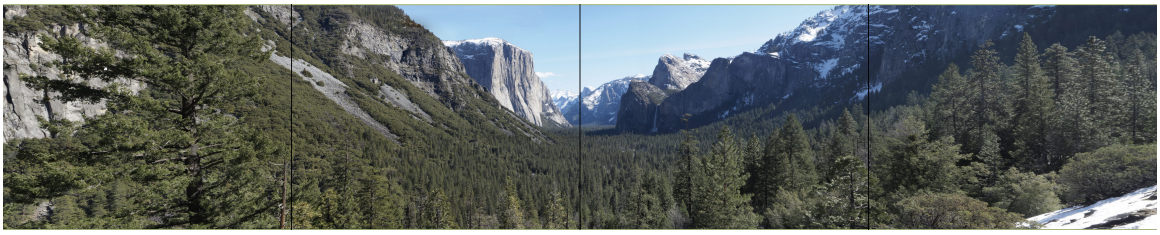


Figure 5.7: Example of a tiled version of the panorama taken at Inspiration Point.

5.2 Online Virtual Field Trip Interface

An online interface has been created that allows one to visually explore the panoramas captured during the field trips via both the Google Earth plugin and a separate image gallery page [2]. It was coded using a combination of the Google Earth API [37], KMLtree [38], jQuery [39], and Highslide JS [40]. This interface could serve as the basis of a platform for making the large amounts of data acquired from the field trips available for other researchers and educators to use for their projects and classrooms.

5.2.1 Google Earth

The GPS measurements taken at each panorama location are plotted as placemarks (read from a KML file) in the map view, which initializes with a top-down view covering the span of locations visited during the field trips (Figure 5.8). In the cases when GPS coordinates were unavailable, an approximate position was selected based on the visual location of the panorama relative to its surroundings and, if possible, other panoramas. A list of these placemarks organized by year is present on the left hand side of the page, above the available "Google Earth Layers" and "Display Options" parameters. To zoom in to a particular placemark location on the map, either double-click that location on the map or double-click next to the placemark name in the list. Once zoomed in to a particular location it becomes easy to explore all of the panoramas taken there by clicking on each of the available placemark icons. When either the name or icon of a placemark is clicked, a pop-up is presented that provides a preview image of that panorama as well as additional information such as the name of the site and its GPS coordinates (Figure 5.9).

Once the pop-up is displayed, clicking on the site name opens a site specific page that contains

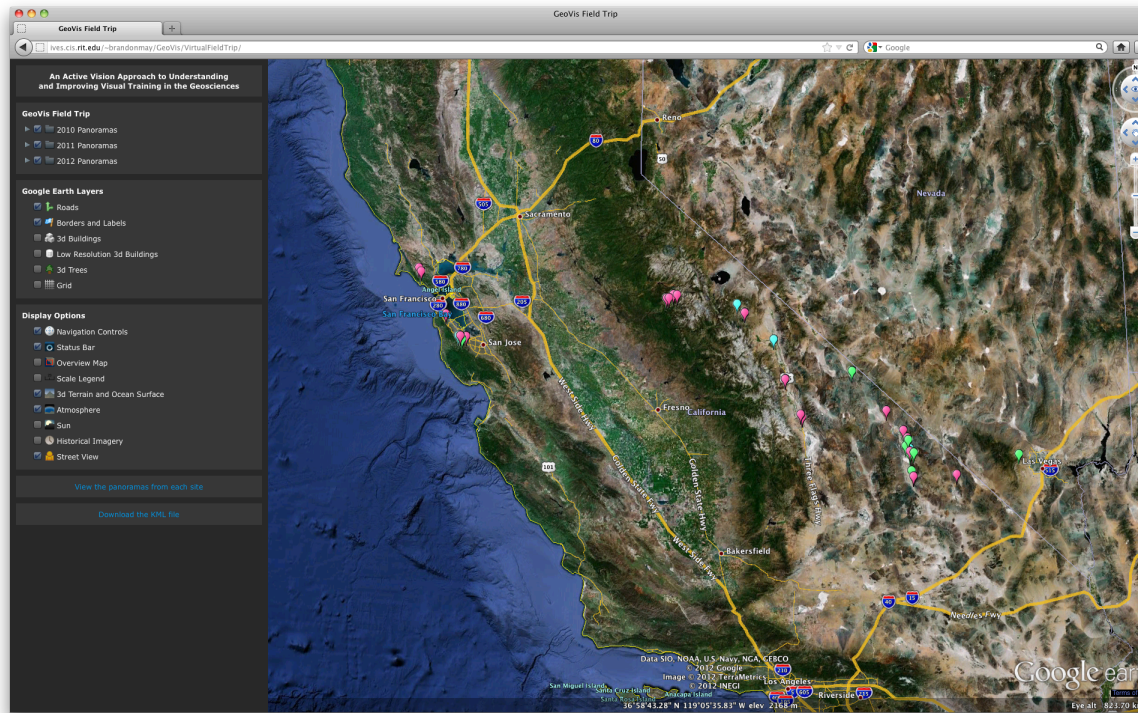


Figure 5.8: A screenshot of the online virtual field trip interface.

all of the panoramas taken for that particular location viewable in a high resolution interactive explorable format [41]. Alternatively, clicking directly on the preview image will open a page containing only that specific panorama in the same interactive format. To download the KML file containing all of the placemark information click on "Download the KML file". The KML file may be opened and explored using the Google Earth desktop application [42].

5.2.2 Site Pages

A specific site page exists for each scene visited during the course of the field trips. Currently they contain a section of interactive explorable panoramas from each year the site was captured. Each panorama has a short description with its name, date of capture, and GPS location (Figure 5.10). These site pages may eventually be expanded to incorporate other field trip components and data such as an overview map of the site, audio and video captured by the mobile eye-trackers, and perhaps some 3D models reconstructed from the image data.

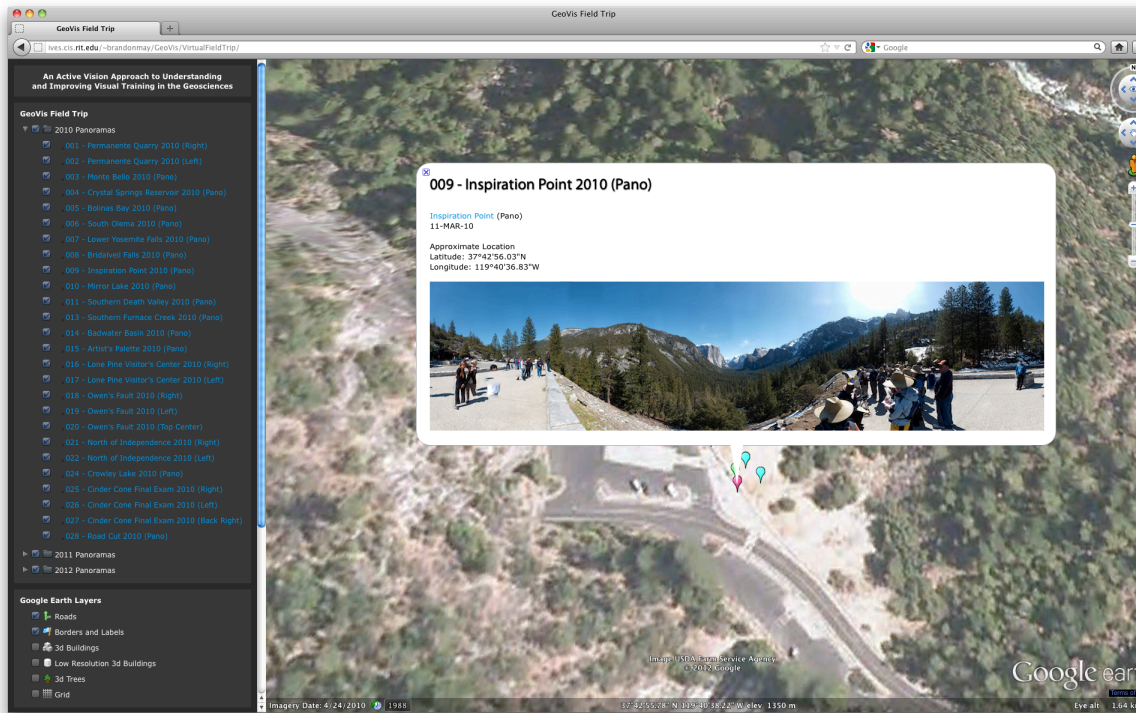


Figure 5.9: A pop-up is presented with additional information when a placemark icon is clicked.

5.2.3 Panorama Gallery

The panorama image gallery page is an alternative to using the Google Earth plugin for exploring the panoramas. This page has an image gallery of the panoramas taken at each site, roughly in the order in which they were visited (Figure 5.11). To access the image gallery page, click on the link "View the panoramas from each site" available underneath the "Display Options". Clicking on any thumbnail image reveals an expanded view with a larger image preview and the same information provided in the Google Earth pop-ups (Figure 5.12). The left and right arrows may be used to browse through each panorama taken at that site. As before, clicking on the name of the site will open the site specific page.

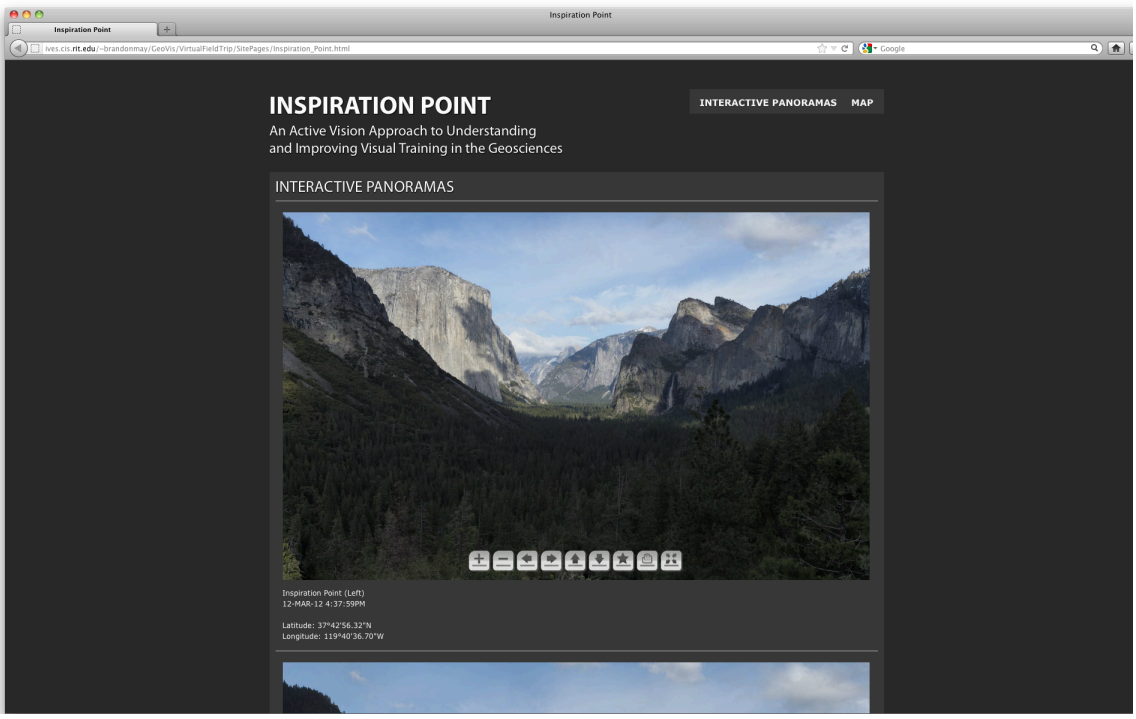


Figure 5.10: A screenshot of the site page for Inspiration Point.

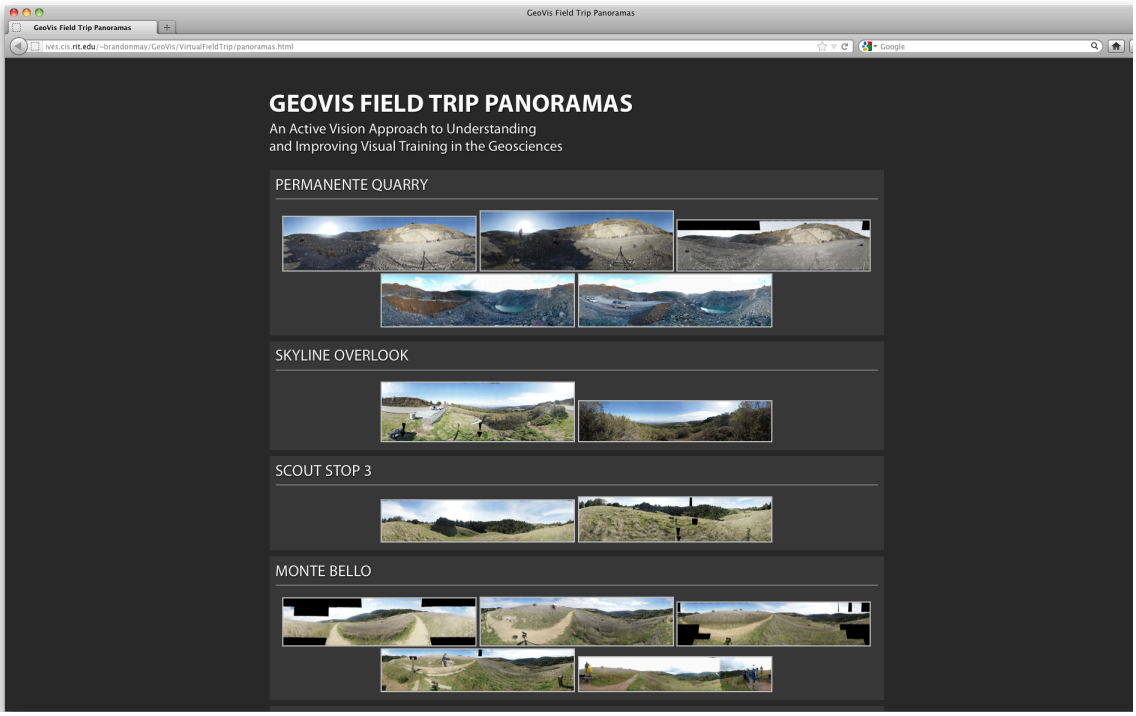


Figure 5.11: A screenshot of the panorama gallery; an alternative to the Google Earth view.

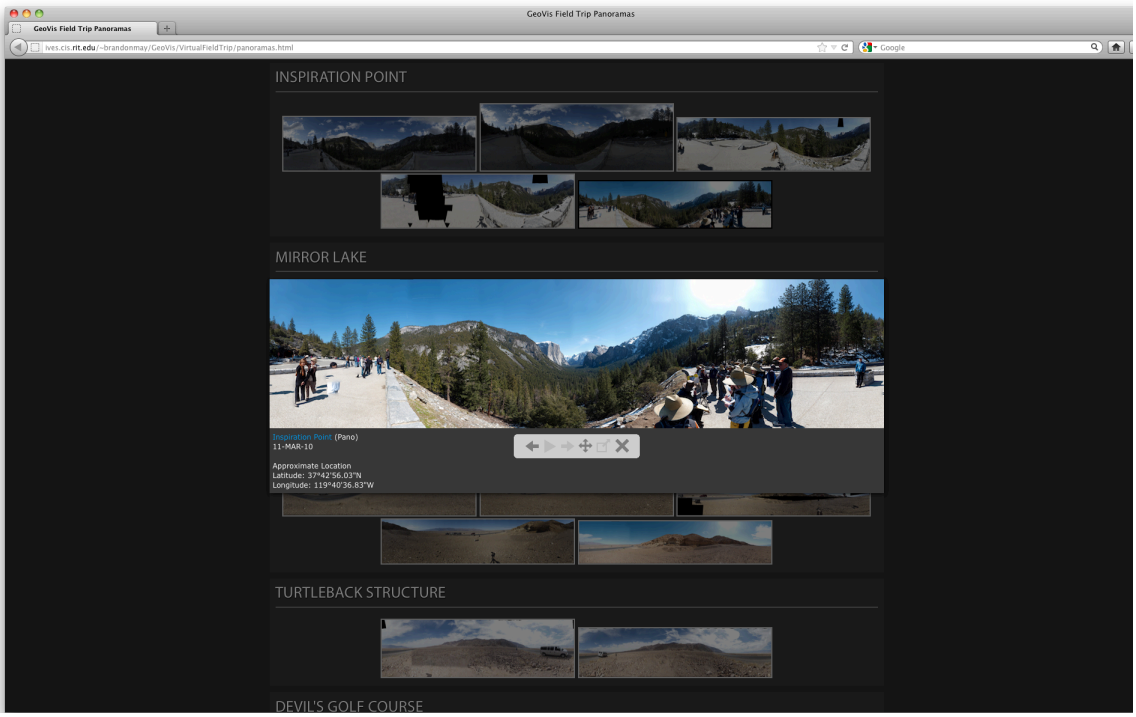


Figure 5.12: Clicking an individual panorama reveals an expanded view with additional information.

Chapter 6

Conclusions

In this work we have developed various imaging hardware and software methods for understanding and improving visual training in the geosciences. For imaging in the field, we built two robotic gigapixel panorama capture systems, one based around the Gigapan and the other based on the Panogear. We designed them to be portable and robust and modified them to allow for extended use and automated image capture. We also developed a shooting pattern for the Panogear that optimized for the greatest amount of coverage in the least amount of pictures. For redundancy, we implemented a dual panorama capture setup at each site that can also be used for stereo image processing. To augment our imagery with location information we put together a GPS measurement methodology that measured vehicle locations & tripod locations using Garmin GPS units that could also be worn while walking around and taking pictures at each site.

To help analyze eye-tracking data, we developed software to aid in the process of registering scene video to panoramas. We implemented a SIFT feature tracking algorithm to generate feature correspondence tracks through a video sequence. We used these tracks to identify keyframes to initially match to the panorama using a combination of SIFT, RANSAC, and a homography model. To match every frame of a video sequence, we used the initial keyframe matches and the video feature correspondence tracks to compute a video frame to panorama homography transformation that varies smoothly from frame to frame. Using this algorithm, we were able to map a set of dynamic control points manually identified in a video sequence to a set of static

control points chosen in a panorama. The mapping errors from the video to the panorama for each valid control point were less than 25 pixels and typically less than 5 pixels.

In order to create a virtual field trip, we implemented a geometric, photometric, and colorimetric algorithm to calibrate a multi-projector curved-screen setup using a single camera. We also developed a projector tiling solution to use as an alternative in the case where the fully calibrated setup was not wanted. To help disseminate our virtual field trip to other educators and the public, we created an online virtual field trip interface using Google Earth. For each geologic site we created a GPS placemark which links to a webpage that has interactive versions of each high resolution panorama captured for that scene.

6.1 Future Work

Beyond the work done for this thesis, there is opportunity for further research in areas such as multiple video-camera structure from motion, collaborative visual SLAM, and multi-view 3D reconstruction.

Instead of relying on a homography model to map scene video to panoramas, a more general structure from motion framework could be used that can compute and track both camera pose and a set of 3D points computed via triangulation to relate gaze to a 3D location in the scene. Because there were several video cameras recording the scenes at the same time from slightly different viewpoints, we have already captured the necessary data for processing through this type of framework. For offline processing, there has been recent research on combining short video segments from multiple handheld video cameras for combined structure from motion computations, which yield a set of accurate camera trajectories and a sparse set of 3D points [43].

For real time processing, such as mapping 3D gaze while the observer is wearing the mobile eye-tracker, a simultaneous localization and mapping (SLAM) approach could be useful. Typically SLAM is performed using a single visual sensor, but recent work has been successful in combining the data from multiple cameras for a more accurate result, called collaborative visual SLAM (CoSLAM) [44]. CoSLAM is able to track the 3D positions of both static and dynamically moving points, while at the same time computing and keeping track of the absolute and relative

pose of each camera used in the computation.

We could also take a more standard and well-tested approach to multi-view 3D reconstruction. Some work has been done in this area focused on the geosciences that uses sets of images taken from various viewpoints of rock samples to reconstruct their 3D geometry [45]. In this same manner, we could capture sets of images similar to these (in addition to the panoramic capture) that would provide the ability to reconstruct other objects in a scene, such as nearby rock walls and various collections of rock samples the students examine. There is a potentially very useful free software package for this purpose called VisualSFM that can take in a set of images or video frames and process them to form a 3D model [46]. It relies on the use of SiftGPU and Multicore Bundle Adjustment for fast geometry processing, and can compute dense 3D by interfacing with PMVS/CMVS [47–49].

Appendix A

Software

During the course of this thesis we have developed several MATLAB algorithms for performing image processing tasks related to registering eye-tracking scene videos to high resolution panoramas. The three main components are:

1. The SIFT feature tracker, which tracks SIFT features through a video sequence and exports a set of feature tracks.

2. The keyframe selector, which decides what video frames are used to anchor the video sequence to the panorama during registration based on finding the local minima of the projective components of the initial homographies.

3. The video frame to panorama registration algorithm, which computes a homography from each video frame in the sequence to the desired panorama using a linear combination of the homographies computed via its two neighboring keyframes.

This code is available for download at:
http://twiki.cis.rit.edu/twiki/bin/view/MVRL/MayCode

References

[1] "An active vision approach to understanding and improving visual training in the geosciences," 2009.

[2] B. B. May. (2012). Geovis virtual field trip, [Online]. Available: http://ives.cis.rit.edu/~brandonmay/GeoVis/VirtualFieldTrip/.

[3] K. Holmqvist, M. Nystrom, R. Andersson, R. Dewhurst, and J. Jarodzka H.and van de Weijer, *Eye Tracking: A Comprehensive Guide to Methods and Measures*. Oxford University Press, 2011, pp. 9–64.

[4] J. Babcock, J. Pelz, and J. Peak, "The wearable eyetracker: a tool for the study of high-level visual tasks," in *Proceedings of the Military Sensing Symposia Specialty Group on Camouflage, Concealment, and Deception*, 2003.

[5] J. Babcock and J. Pelz, "Building a lightweight eyetracking headgear," in *ETRA 2004: Eye Tracking Research and Applications Symposium*, 2004, pp. 109–113.

[6] N. Snavely, S. Seitz, and R. Szeliski, "Photo tourism: exploring photo collections in 3d," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 835–846, 2006.

[7] M. Brown and D. Lowe, "Unsupervised 3d object recognition and reconstruction in unordered datasets," in *International Conference on 3D Imaging and Modelling*, Nice, France, 2003, pp. 1218–1225.

[8] ——, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.

[9] D. Steedly, C. Pal, and R. Szeliski, "Efficiently registering video into panoramic mosaics," in *IEEE International Conference on Computer Vision*, vol. 2, 2005, 1300–1307 Vol. 2.

[10] R. Szeliski, "Image Alignment and Stitching: A Tutorial," *Foundations and Trends in Computer Graphics and Vision*, vol. 2, no. 1, pp. 1–104, 2006.

[11] J. Shi and C. Tomasi, "Good features to track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94))*, Seattle, WA, 1994, pp. 593–600.

[12] C. Schmid, R. Mohr, and C. Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision*, vol. 37, no. 2, pp. 151–172, 2000.

[13] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2003.

[14] T. Tuytelaars and K. Mikolajczyk, "Local Invariant Feature Detectors: A Survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2007.

[15] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[16] A. Vedaldi and B. Fulkerson. (2008). VLFeat: an open and portable library of computer vision algorithms, [Online]. Available: http://www.vlfeat.org/.

[17] ——, "VLFeat: An open and portable library of computer vision algorithms," 2010, pp. 1469–1472.

[18] J. M. Morel and G. Yu, "Is SIFT scale invariant?," *Inverse Problems and Imaging*, vol. 5, no. 1, pp. 115–136, 2011.

[19] R. Szeliski, *Computer Vision Algorithms and Applications*. Springer, 2011.

[20] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2008.

[22] R. Raskar, J. van Baar, T. Willwacher, and S. Rao, "Quadric transfer for immersive curved screen displays," in *Eurographics*, vol. 23, 2004.

[23] R. Raskar, M. S. Brown, R. Yang, W. C. Chen, G. Welch, H. Towles, B. Scales, and H. Fuchs, "Multi-projector displays using camera-based registration," in *Visualization*, 1999, pp. 161–522.

[24] M Brown, A Majumder, and R Yang, "Camera-based calibration techniques for seamless multiprojector displays," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 2, pp. 193–206, 2005.

[25] G. Wallace, H. Chen, and K. Li, "Color gamut matching for tiled display walls," in *Workshop on Virtual environments*, 2003, pp. 293–302.

[26] A Majumder, Z He, H Towles, and G Welch, "Achieving color uniformity across multi-projector displays," in *Visualization*, 2000, pp. 117–124.

[27] B. Sajadi and A. Majumder, "Auto-calibration of cylindrical multi-projector systems," *IEEE Virtual Reality Conference (VR)*, pp. 155–162, 2010.

[28] B. Sajadi, M. Lazarov, M. Gopi, and A. Majumder, "Color seamlessness in multi-projector displays using constrained gamut morphing," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1317–1326, 2009.

[29] (2010). Gigapan systems, [Online]. Available: http://www.gigapan.com/cms/about-us.

[30] Kolor. (2011). Panogear: the motorized panoramic head | kolor, [Online]. Available: http://www.kolor.com/panogear-motorized-panoramic-head.html.

[31] ——, (2011). Papywizard, the app driving the kolor panogear head via bluetooth, [Online]. Available: http://www.kolor.com/panogear-motorized-panoramic-head-download.html.

[32] J.-Y. Bouguet. (2010). Camera calibration toolbox for MATLAB, [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/.

[33] Kolor. (2010). Autopano giga image-stitching software, [Online]. Available: http://www.kolor.com/image-stitching-software-autopano-giga.html.

[34] P. D. Kovesi. (2000). MATLAB and Octave functions for computer vision and image processing, Centre for Exploration Targeting, School of Earth and Environment, The University of Western Australia, [Online]. Available: http://www.csse.uwa.edu.au/~pk/research/matlabfns/.

[35] O. Chum, T. Pajdla, and P. Sturm, "The geometric error for homographies," *Computer Vision and Image Understanding*, vol. 97, no. 1, pp. 86–102, Jan. 2005.

[36] A Majumder and R Stevens, "Perceptual photometric seamlessness in projection-based tiled displays," *ACM Transactions on Graphics (TOG)*, vol. 24, no. 1, pp. 118–139, 2005.

[37] Google, Inc. (2012). Google earth API, [Online]. Available: https://developers.google.com/earth/.

[38] underbluewaters. (2012). Kml tree navigation widget for use with the google earth api using jquery, [Online]. Available: http://code.google.com/p/kmltree/.

[39] The jQuery Foundation. (2012). Jquery: the write less, do more, javascript library, [Online]. Available: http://jquery.com.

[40] Highsoft Solutions AS. (2012). Highslide js - javascript thumbnail viewer, [Online]. Available: http://highslide.com.

[41] krpano Gesellschaft mbH. (2012). Krpano panorama viewer, [Online]. Available: http://krpano.com.

[42] Google, Inc. (2012). Google earth (version 7.0.2.8415), [Online]. Available: http://www.google.com/earth/index.html.

[43] G. Zhang, H. Liu, Z. Dong, J. Jia, T.-T. Wong, and H. Bao, "Efficient Non-Consecutive Feature Tracking for Large-Scale Structure-from-Motion," *International Journal of Computer Vision*, pp. 1–19, Nov. 2011.

[44] D. Zou and P. Tan, "Coslam: collaborative visual slam in dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 2, 2013.

[45] M. Favalli, A. Fornaciai, I. Isola, S. Tarquini, and L. Nannipieri, "Multiview 3D reconstruction in geosciences," *Computers & Geosciences*, 2011.

[46] C. Wu. (2012). Visualsfm: a visual structure from motion system, [Online]. Available: http://homes.cs.washington.edu/~ccwu/vsfm/.

[47] ——, (2012). Siftgpu: a gpu implementation of sift, [Online]. Available: http://www.cs.unc.edu/~ccwu/siftgpu/.

[48] ——, (2012). Multicore bundle adjustment, [Online]. Available: http://grail.cs.washington.edu/projects/mcba/.

[49] Y. Furukawa. (2011). Clustering views for multi-view stereo (cmvs), [Online]. Available: http://grail.cs.washington.edu/software/cmvs/.